2

AGARD-CP-417

AGARD-CP-417

AD-A198 666

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

AGARD CONFERENCE PROCEEDINGS No.417

The Design, Development and Testing of Complex Avionics Systems

NORTH ATLANTIC TREATY ORGANIZATION

DISTRIBUTION AND AVAILABILITY
ON BACK COVER

88 4 11 307

②

AGARD-CP-417

NORTH ATLANTIC TREATY ORGANIZATION

ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT

(ORGANISATION DU TRAITE DE L'ATLANTIQUE NORD)

AGARD Conference Proceedings No.417

THE DESIGN, DEVELOPMENT AND TESTING OF COMPLEX AVIONICS SYSTEMS

Papers presented at the Avionics Panel Symposium held in Las Vegas, US,
on 27 April—1 May 1987.

THE MISSION OF AGARD

According to its Charter, the mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community;
- Providing scientific and technical advice and assistance to the Military Committee in the field of aerospace research and development (with particular regard to its military application);
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Exchange of scientific and technical information;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced directly from material supplied by AGARD or the authors.

Published December 1987

ISBN 92-835-0437-2

Printed by Specialised Printing Services Limited
40 Chigwell Lane, Loughton, Essex IG10 3TZ

THEME

This symposium was designed to explore how today's system designer is addressing the solution to tomorrow's avionics systems design.

As government budgets become more limited for the research, development, testing and production of military aircraft systems, and the Warsaw Pact nations continue to produce all types of aircraft in greater numbers, the NATO nations must "re-look" at how avionic systems are developed. In general, the avionics community has thought of avionic architecture as the integration of a collection of "black boxes" (sensors, navigation, communication, displays, etc.) with the software allowing for the communication between "black boxes", computers and man. Normally, the system is decomposed into manageable parts with accurately defined interfaces. By rigidly controlling this process, aerospace companies have developed excellent avionics systems which are fully integrated into aircraft systems, but the cost is high. Cost means that the aircraft is required to perform multi-missions which often lead to performance compromises. With the advent of the VHSIC, distributed processing, artificial intelligence, sensor data fusion, etc., the technologists are blurring the clear functional allocation defined for the "black box". These factors, technology and cost, provide both an opportunity and a challenge to the system designers to design future avionics systems whose performance degrades gracefully, is reliable, and is affordable.

***

Ce Symposium avait pour thème les méthodes utilisées aujourd'hui par un concepteur de système pour trouver une solution aux problèmes posés par les systèmes avioniques de demain.

Comme les gouvernements limitent de plus en plus les budgets consacrés à la recherche, au développement, aux essais et à la production des systèmes d'avions militaires et comme les pays du Pact de Varsovie produisent de plus en plus d'avions, les pays de l'OTAN doivent reconsidérer les méthodes de développement des systèmes avioniques. D'une manière générale, l'industrie avionique considère l'architecture d'une système avionique comme un ensemble de "boites noires" (détecteurs, navigation, communication, affichage, et...), le système se décompose en éléments pouvant être gérés et ayant des interfaces définies avec précision. Par une maîtrise stricte de ce processus, les sociétés aérospatiales ont développé d'excellents systèmes avioniques qui sont totalement intégrés dans les systèmes de l'avion, mais dont le coût est élevé. Pour des raisons de coût, l'avion doit pouvoir effecteur des missions multiples, ce qui nécessite des compromis entre les performances. Avec l'apparition des circuits intégrés à très grande échelle (VHSIC), de l'intelligence artificielle, de la fusion des données de détecteur, etc...., les technologies brouillent l'image claire que l'on avait de l'affectation fonctionnelle définie pour la boîte noire. Ces facteurs techniques et économiques offrent une occasion et un défi à relever pour les concepteurs de système; ceux-ci doivent concevoir les futurs systèmes avioniques dont le coût et la fiabilité doivent être satisfaisants et dont les performances ne doivent se dégrader que lentement.

AVIONICS PANEL

Chairman: Dr G.H.Hunt
ADXR (E)
Royal Aircraft Establishment
Farnborough, Hants, GU14 6TD
UK

Deputy Chairman: Dr R.W.MacPherson
National Defence Establishment
CRAD/DS Pol3
101 Colonel By Drive
Ottawa, Ontario, K1A 0K2
Canada

TECHNICAL PROGRAMME COMMITTEE

PROGRAMME CO-CHAIRMEN

Mr E.Yanuzzi
Director, Software & Computer Dir.
Naval Air Development Center
Warminster, PA 18974–5000
USA

Dr E.Kutchma
Head, Aircraft Weapons Integr. Dept.
(Code 31)
US Naval Weapons Center
China Lake, CA 93555
USA

COMMITTEE MEMBERS

Mr R.Guiot
Avions Marcel Dassault-Breguet Aviation
Direction Générale Technique
Division Systèmes d'Armes
78, Quai Carnot, BP 300
92214 Saint Cloud Cedex, France

Ing. L.Crovella
Aeritalia SAIPA
Gruppo Sistemi Avionici ed
Equipaggiamenti
10072 Caselle Torinese
Torino
Italy

Maj. M.Mascarenhas
Head, Test Software Group
O.G.M.A.
2615 Alverca, Portugal

LTC. W.Van den Branden
Belgian Air Staff VDT/B
Section Avionics
Quartier Reine Elisabeth, Rue d'Evere 1
1140 Bruxelles, Belgium

Mr J.Whalley
BAe Aircraft Group
Weybridge Division
Chester Road, Woodford, Bramhall,
Stockport, Cheshire, SK7 1QR
UK

Dipl. Ing. W.Küny
MBB Dep LKE3
Postfach 80 11 60
8000 München 80
Federal Republic of Germany

AVIONICS PANEL EXECUTIVE

Lt. Colonel M.Stratton

From Europe
AGARD-OTAN
7, rue Ancelle
92200, Neuilly-sur-Seine
France
Tel: 47.38.57.65
Telex: 610176 AGARD

From the USA and Canada
AGARD-NATO
APO New York 09777

CONTENTS

*Abstract only. The full text appears in Classified Publication CP 417 (Supplement)

TECHNICAL EVALUATION REPORT ON THE 53RD SYMPOSIUM OF THE AVIONICS PANEL OF AGARD

THE DESIGN, DEVELOPMENT, AND TESTING OF COMPLEX AVIONICS SYSTEMS

Las Vegas, Nevada, U.S., 27 April - 1 May 1987

David Schimsky
Naval Air Development Center
Warminster, Pennsylvania, U.S. 18974

SUMMARY

The overall quality of the papers presented at the symposium was first class, and the presentations were, on the whole, excellent. However, the relevance of the subject matter to the stated topic of a session was weak in several cases. Although these papers addressed the primary topics in the broadest sense, their subject matter was often unfocused and diffused. Their primary point was usually a reiteration of how avionic system design, development, and test was accomplished on a specific project. It would have been better if these papers clearly stated the problem and how it was solved. If we in the avionics community had more standardized ways of doing business, we would most likely face the same problems, and any discussion on their resolution would certainly draw our attention.

The symposium clearly emphasized two vital requirements in the development of avionic systems:

- Current and future avionic systems are so complex and require such a diversity of talents, expertise, and resources to design, develop, and test that a stringent, rigorous, methodical top-down approach using the latest computer-aided techniques must be applied.

- Standards are needed in almost every area.

Without some sort of agreement and implementation of standards to impose discipline, future symposia will repeat the theme of this one: "Here is how I did it, now show me how you did it." While *exploring different approaches and sharing* lessons learned are beneficial, more progress could be made by at least agreeing to use the same approaches, techniques, etc., and evaluating our successes, failures, and problems when the next symposium is convened.

THEME AND OBJECTIVES

The title of this symposium was "The Design, Development, and Testing of Complex Avionics Systems." However, this title barely does justice to the subject. In the recent past, a near revolution has occurred in avionic system requirements, including how they are built and, as was continuously emphasized in the presentations, how they are operated.

The stated theme was "to explore how today's system designer is addressing the solution to tomorrow's avionics system design." The presentations adequately addressed this theme with descriptions of current and near-future efforts, all with forward-looking implications.

It was clear from the content of the presentations that all member countries, as well as the companies within those countries, are experiencing the same problems in designing advanced avionic systems. What is truly remarkable is the virtual unanimity in the approach to resolving the problems.

The symposium's goal was to share the experiences, successes, and pitfalls surrounding this complex subject and to take advantage of lessons learned. Judging from the size of the audience at each session, the session, the enthusiastic question and answer periods following each presentation, and the variety of experiences discussed, the symposium may be considered a resounding success.

TECHNICAL CONTENT

The symposium began with an opening address by VADM G. Clarke, Commander of the Space and Naval Warfare (SPAWAR) Systems Command of the United States Navy. He set the tone for the meeting by strongly emphasizing the need for system designers to consider the whole system in the design process rather than the avionics portion only. According to the Admiral, the whole system includes surface ships, aircraft, submarines, space vehicles, and, most importantly, man. Admiral Clarke then described his task as Commander of SPAWAR to establish a total system architecture approach to the battle group and to do it top-down for the first time ever. He described the process he and his staff are using and the three major components of the architecture: the tactical command system, the communication support system, and the warfare support system.

Admiral Clarke's comments were reiterated throughout the symposium. His emphasis on an orderly, rigorous, structured top-down approach was the single most common element of virtually all presentations.

Session 1 — Design Aspects of Future Avionic Systems

In Paper No. 1,* the author describes a cooperative planning effort between the U.S. National Aeronautics and Space Administration (NASA) and the Department of Defense (DOD) to conduct studies to determine technology development requirements for the design of the next generation of space transportation systems. A thorough top-down hierarchical approach being used on this project is described. This approach covers major categories such as guidance, navigation, and control; flight systems management; system integration; and modeling, communications, and man systems interfaces. The multitude of research elements that must be considered provides ample justification for using a structured, rigorous approach. The paper highlights the need for significant systems as an important driving factor. While the paper addresses only one of five major phases of the avionic system planning process, a systematic, classical approach is taken: Define the potential (generic) mission, define the (generic) vehicle to satisfy the mission, and identify the technology needed to build the vehicle. The time frame for the first vehicle is post-2000.

Paper No. 2 calls for standardization at various levels to help solve the problem of transitioning today's architectures into architectures of the future without having to start from the beginning. As the author points out, we never start with a totally new design, but one that is almost always based upon an existing design. Therefore, a "bridging" or transitioning technique must be developed. He identifies two requirements to accomplish this:

- A doctrine that defines both future architectures and the intermediate "transitional" architectures.

- An interplatform interface requirements document.

The author proposes to develop a high-level architecture of the future and a standard method to identify the components and structure of that architecture. He also calls for some form of standardization or commonality across platforms, and even across national boundaries, a view that was expressed by many throughout the symposium.

Paper No. 3 highlights the role of sensors in a hypothetical future aircraft and emphasizes the complex problems facing the pilot. These aircraft will be equipped with highly capable sensors with different data rates, ranges, and accuracies that sometimes do not supply data (e.g., because of jamming). The analysis, reduction, fusion, correlation, and tracking of these data can be accomplished in a decentralized, adaptable system with the aid of artificial intelligence (AI), also known as the "pilot's associate." The author states that the successful developer of the future will be the one that best manages the myriad of different contractors that will be required to design and integrate such a system.

Papers Nos. 4 and 5 introduce advanced tools to aid the design of complex systems. Expert Consultant for Avionic Systems Transformation Exploitation (ECATE), described in Paper No. 4, is an expert system that aids the rapid prototyping of avionic systems. Paper No. 5 discusses the Controlled Requirements Expression (CORE) system, which is useful in providing unambiguous functional descriptions and representing dynamic interactions within and across systems. The use of sophisticated tools will be mandatory to ensure the successful execution of current and future complex avionic system designs. This issue raises two questions:

- Will the ability to design more complex systems depend on the ability to design increasingly complex tools?

- Can tools be used to design tools?

Work on the Experimental Aircraft Program (EAP) is the topic of Paper No. 6. The author describes the use of the CORE tool to impose a top-down structured design approach. The use of CORE resulted in the remarkable conclusion that after 20 flights of the aircraft in the first 18 days of flight, no system changes were deemed necessary. Apparently, enforcing a superior, strictly followed approach can lead to outstanding results. This raises the suspicion that many problems encountered in avionic system design projects may be due to taking shortcuts and known high-risk paths.

Paper No. 7 best addresses the subject of the first session. This paper describes the development of a generic architecture of the future. The author identifies the salient characteristics of the future system, such as distributed processing and control (not shared processing), fault tolerance leading to uninterrupted operation, and stringent real-time performance demanding very short control loops. The paper also describes a very practical, professional approach taken to these subjects and discusses the problems of real-time processing. The author also calls for some form of standardization in control/data distribution to facilitate the task of system designers, in this case the Society of Automotive Engineers (SAE).

*A list of the complete title and authors of each paper is presented at the end of this paper.

The use of special test "rigs" and predefined standard interfaces to expedite the testing of the EH-101 helicopter integrated avionic suite is discussed in Paper No. 8. A "rig" is actually a complex system in itself, consisting of computers, emulators, simulators, buses, etc., that allow testing to proceed without the delays associated with the lack of some system components. Again, initial planning and standardized interfaces helped the system designers considerably in the latter development stages.

Paper No. 9 describes the use of the formalized System Engineering Technique (SET), which was developed to improve both the quality and productivity of system engineering tasks. The authors emphasize that no tool, technique, or methodology will replace common sense, and that SET is not really new, but a formalization and integration of existing procedures and worksheets. The authors describe the use of three models:

- A functional model that outputs requirements.
- A physical model that is represented by the design.
- An operational model used for final system analysis.

These models should lead to specific quantifiable constraints and system components. This particular paper generated much interest among the symposium attendees, as evidenced by the large number of pertinent questions.

As described in Paper No. 10, a rigorous, strictly adhered to methodology was responsible for the successful development of the RAFALE aircraft. Rapid prototyping of complex systems was also beneficial. This technique improved the quality of the functional design specifications, which, as the author points out, are the cornerstones of the whole effort. Rapid prototyping was used to validate the functional specifications that were prepared by adhering to a rigid top-down methodology. As a result of these procedures, a first flight was conducted six months ahead of schedule, and the number of errors in the specifications was greatly reduced.

Paper No. 11 is a thoughtful presentation on the impact of software on avionic architecture. A brief review of avionic system development history shows that technology, as a whole, has grown exponentially, with hardware following in a very similar manner. However, software capabilities and man's ability to produce it have, unfortunately, not followed suit. Yet, software will be the main constraint in satisfying future requirements; no alternatives are foreseen. A new model was proposed, wherein the rework required after an error is found will be limited to a small step backward in time rather than the large rework loop that is used on most of today's efforts. The author feels that careful definition of information flows early in the design process is the key.

The 12th paper listed on the agenda was not presented at the symposium.

In Paper No. 13, the author compares integrated and separate systems for flight control and navigation. The main problem is that the two subsystems have somewhat divergent characteristics. The problems of vibration, data processing, and increased vulnerability as a result of separating the systems are discussed. The author concludes that the integration of flight control sensors and the inertial navigation system (INS) in one system is basically feasible and has been successfully implemented.

Paper No. 14 emphasizes that while new technologies provide designers with all sorts of new opportunities, they are also posing significant challenges. The sheer complexity of the proposed systems demands that a stringent, predictive, analytical methodology be employed. Further, the role of man in the system must be given the attention it is due. The author believes that man is, after all, at least as important a component as, for example, a radar. Man's performance in the system should be analyzed in as much detail as the system hardware. The paper describes an approach that encompasses mission functional requirements analysis, candidate system development and gross task analysis, critical task analysis, system performance analysis, and validation. Again, a rigorous, analytical top-down approach seems to be the key to success.

In paper No. 15, the authors discuss the use of a formalized, structured methodology called "Crewstation Information and Development System" (CIDS). The importance of specific requirements that stand on their own merit is stressed, and they are used throughout CIDS. CIDS provides a quantified method for making critical design decisions, and, therefore, is suitable for partial or complete automation. As a result of using CIDS, the design can be considered optimal in accordance with the parameters and weighting factors chosen by the designers. The system provides requirements traceability, degraded mode handling, redundancy handling, and strict interface designs. A concept demonstration of CIDS is due in the fourth quarter of 1987.

Paper No. 16, delivered by R. DeSipio, addresses the role of man in the system. The increase in raw data to the human in the system will overload him. Data must be converted into processed information before they are presented to the human for decision making. Even the decision-making process should be augmented by knowledge-based systems, if he is to devote the majority of his attention to the tactical situation. The authors propose a reorientation of the system design process, in which the design is based

upon decision requirements rather than hardware performance. It appears that the more complex the system, the harder the job for the man in the loop. Or, as one participant put it, "high technology equals high workload."

Session 2 — Managing the Future System Design Process

In the first paper of the second session, Paper No. 17, the author addresses a tool that Northrop uses as an aid in managing avionic system design. The Avionic System Engineering Tool (ASET) automates the company's structured design approach, which comprises four phases:

- Abstract requirements identification.
- Requirements and functional decomposition.
- Functional recomposition.
- Detailed interface and bus definition.

ASET was designed to help diseminate information; be expandable, maintainable, fast, powerful, and user-friendly; require only a short learning curve; produce hard copy; and handle classified data. As overwhelming as these requirements appear, the author claims that ASET meets them. Further, it provides traceability, modification time decreases, I/O verification, throughput and memory analyses, and improved software designs and test.

Paper No. 18 addresses issues concerning the human in the cockpit, ergonomics, and the use of AI. The authors describe two experiments under way in the laboratory, one dealing with vision and imaging, the other with cognitive psychology and cognitive aids in cockpits. To date, databases have been produced on the transfer functions of vision and the results of peripheral vision studies. These databases have been used in developing models of vision to aid interpretation of complex images. The process is now being used to interpret satellite image data. This knowledge will aid pilots under stresses of acceleration and vibration and reduce the effects of hypoxia.

The objective of the second experiment, which involves cognitive psychology, is to assess the use of AI, working in concert with the pilot in real-time, to provide an analysis of the pilot's current situation. The "context detection unit" is especially interesting and appears to be a rather advanced application of AI.

Paper No. 19 presents an approach to automated cockpit design with a tool called "Cockpit Automation Design Support System" (CADSS). The authors identify a number of deficiencies associated with today's way of doing business, including lack of standardization, dependence on people's unique abilities, manual procedures, outdated design guides, and poor change management. While the approach taken by these authors is similar to others described at the symposium, CADSS is uniquely tailored for and applied to cockpits. Competing teams are building CADSS now, and demonstrations are scheduled for May 1988.

The fourth paper of this session, Paper No. 20, describes a system for search and rescue on a helicopter. The paper identifies the eight-leg mission used as a scenario and the three subsystems constituting the architecture: the display, the autopilot/navigation subsystem, and mission management subsystem. A detailed technical explanation of the navigation and mission management subsystems revealed a complex, capable equipment suite. The Nadir Mk2 computer, for example, is a 32-bit microprogrammable system that can execute at a rate of 1 MOPS. The system was specifically designed for the type of real-time operations needed for this project. The development effort required 3 1/2 years to complete (7 months for definition), and the first flight occurred in 18 months. Software development was performed on a VAX 785 and a MICROVAX, with the aid of microprocessor emulators.

The authors describe a complex interactive relationship that exists between the system designers/integrators and the equipment manufacturers. High rates of information flow between the two groups, and complementary tools aid successful interaction. This aspect of system design was not touched upon in other papers, but it is certainly worthy of further consideration.

In Paper No. 21, the author effectively justifies the need for high levels of automation and fully integrated architectures with powerful central processing capability. His description of a helicopter communication system and the concerns faced by system designers in building the system highlight some of today's problems. Nap-of-the-earth communication, target data handoffs, auto reconfiguration upon failure, and operations in the presence of intense jamming are enough to convince even the most skeptical that ad hoc methods are grossly outdated.

Paper No. 22 focuses on one of the main problems addressed at this symposium; that is, the design of crew systems has typically involved little systematic consideration of human performance characteristics and limitations. The author describes an Air Force thrust to manage design information, namely the Integrated Perceptual Information for Designers (IPID) Project. The goal of the IPID Project is to consolidate human performance data, present these data in useful formats, train designers in the use of the

data, and make the data accessible. Descriptions are provided on how the project personnel intend to meet the goals of the IPID Project. It seems obvious that some approach, such as the one described in the paper, must be used if the more-or-less heuristic approach to design is to become more scientific.

In Paper No. 23, the author describes a project to produce interchangeable modules (hardware) manufactured by two different contractors. The statement of work mandated the exchange of modules of equivalent function without any impact on software. The MIL-STD-1750A computer was produced out of a VAMP module set. The illustrations show that two very dissimilar-looking modules are actually replacements for each other. At the end of the project, interoperability will be demonstrated. The author also declares that the biggest problem associated with using standard hardware is the software and predicted this problem will become worse.

Paper No. 24 describes the problems electromagnetic fields cause modern electronic devices. All the characteristics that are inherent to modern high-technology devices, such as the change from black boxes to integrated devices, the use of low-voltage circuitry, and the use of VHSIC-sized components, make it easier for stray, unwanted electromagnetic fields to upset the state of the devices. The author calls for guidelines in many areas, including grounding, bonding, filtering, shielding, electrical interfaces, etc. Further, to be useful in practical applications, these guidelines must be specific, applications, not generic "motherhood" statements. Testing also needs improvement. In accordance with MIL-STD-461/462A/B, current methods imply stand-alone equipment designs, not full systems. In addition, susceptibility criteria are not clearly defined.

The integration and testing of an airborne radar is discussed in Paper No. 25. This paper provides some unassailable advice: An informed, fundamental, and methodical approach is needed; software integration is important; tests must be performed on the actual radar and not only on the simulator; and an acceptance test may be required for the sponsor. While most of the material presented is commonly accepted, some of the

The final paper of this session, Paper No. 26, describes expected developments in microelectronics technology in the next 10 to 15 years. The author points out that in the last 20 years, microelectronic devices have increased 200 percent in density and 20 percent in speed. The future will probably bring CMOS, lower power consumption, higher densities, and submicron dimension. By the year 2000, 0.3-micron feature sizes will be possible. The author also explains different kinds of IC processing and technologies and declares that perhaps the best of two worlds might be achieved by a BIMOS technology (e.g., CMOS/BIPOLAR). The message of this paper is to expect rapid advances, including some that were thought impossible just a few years ago, and more computer tools.

Session 3 - System Design Tools and Integration

In the first paper of the third session, Paper No. 27, the author uses convincing historical data from 10 projects over the last 12 to 13 years to show that the projects that used consistent and in-depth human engineering processes had the fewest design changes. One of the points made in this paper was voiced earlier in the symposium at least twice: Look ahead, perceive how equipment will be used in conjunction with the human being, and design functionality into the equipment.

Large-scale software development and testing are the focus of Paper No. 28. The most advanced, state-of-the-art tools, approaches, and methodologies are described. A typical cross-development is described. The software validation rack provides environmental simulation, graphic output, and various operational modes. The author identifies a trend that can be readily perceived: Front-end tools (e.g., specification tools) will be integrated with back-end tools (e.g., test tools). A natural progression from specification tool, to semiformal specification, to prototype, to stimuli, to test tool is foreseen. The potential for massive automation of the process seems obvious.

Paper No. 29 describes the use of CAD/CAM technology as it can be applied to the system design process.

In Paper No. 30, the author descries current design automation systems' almost total lack of ability to capture and communicate the context within which a design is developed. He points out that current standardization efforts focus on syntax, do not separate "what" from "how," and use ad hoc semantics, among other faults. He describes a process called "Abstract Resource Design Methodology," which may have the potential to correct many of the stated deficiencies. These complaints are valid, although the impact of the deficiencies on the concept-design-implementation process is not obvious.

Paper No. 31 describes how the Avionics System Test Training Aircraft (ASTTA) can be beneficial in the design process. Because designers have limited viewpoints about the operational environment and testers generally lack avionic system test training, the resulting system can often overtax and overwhelm system users. A flying test bed employed early in the process can alleviate some of these problems. The audience tended to challenge the concept's nonreal-time operation and its less than 100-percent system fidelity. The question of cost-effectiveness was also raised.

Paper No. 32 addresses a modern, high-powered approach to software engineering used by British Aerospace (BAE) on the Experimental Aircraft Program (EAP). Modern tools, such as CORE, PERSPECTIVE, and DASS, and capable host computers (i.e., VAXs) were used in concert with sound engineering practices (e.g., thread diagrams, automated documentation, etc.) to increase productivity and reduce errors. The author quotes figures of a five-fold increase in productivity and a five-fold reduction of errors (from 10 to 2 errors per 1,000 LOC). Again, the author notes a trend toward true integration of tools and methodologies.

Paper No. 33 discusses the development of a set of automated tools to aid in the design of avionic systems, from the definition phase through the software production phase. (The tools are actually described in some detail in Paper No. 36, which was presented immediately following this paper.) The author describes the by-now familiar characteristics of the systems we are concerned with: highly integrated, open-ended, and safe. The major problems in designing such a system successfully (i.e., one that meets requirements within budget and on time) are cost/schedule control, communications between the different contractors, validation of the system, and management of the massive amount of documentation that is produced. A thorough description of the steps involved in the various design phases is also provided in this paper.

A software life-cycle environment being built for the Rome Air Development Center is discussed in Paper No. 34. This VAX-based, multilingual (JOVIAL, FORTRAN, Ada, COBOL), multirole, distributed facility covers the full software life cycle. The authors point out that 32 tools will provide users with basic capabilities in requirements definition, design, prototyping, coding, test, verification, project management, configuration management, environment management, etc. Full functional capability is scheduled for August 1988. The audience questioned the portability of tools developed for a virtual memory system (VMS).

Paper No. 35 describes a system to be installed in a research aircraft that comprises 11 subsystems. The system will be used to help solve the problems of aircraft routing caused by increased air traffic that must be handled by a constant number of airports. The author believes that new avionic systems (e.g., microwave landing systems, Global Positioning System, electronic flight instrument systems, etc.) can help alleviate the situation if properly used. The research aircraft will be used as a tool and should be finished by the end of 1999.

Paper No. 36 is a "companion" paper to Paper No. 33, both of which were presented in the third session. In this paper, the authors expand upon the avionic development system that was described in Paper No. 33. The authors explain that this concept, developed by/for the French avionic community, resulted in an open-ended suite of integrated tools that is useful in system design and software development. Each participating contractor may add on or tailor the basic tool set to his individual requirements. The authors describe a "hosting structure" tool that ties the whole system together and three other basic tools: OCS, a system design aid; DLAO, a computer-aided software definition tool; and SAO, a graphic, detailed specification language.

The authors also indicate that some of the tools developed have already been used on some recent French projects, and that the workshop concept, supported in part by the French Ministry of Defense, was successful in aiding the definition of complex avionic systems, the production of better specifications, and, perhaps most significantly, the communications between the various entities involved in the effort.

A "coherent" functional development methodology is the focus of Paper No. 37. This concept was developed over the past five years and is used by Rockwell Aviation. The term "coherent" identifies the need to take into account the interactions of the real system, on a time-line basis, to produce an effective, workable system. The technique and tools to develop such a system are embodied in a methodology called "Coherent Design Evaluation Simulation" (CODES), which is used to produce high-quality, real-time, man-in-the-loop simulations. The tool couples the man with avionic, vehicle, and flight control models and imposes a time-line scenario. The power, complexity, and sophistication of CODES are evident from its statistics: hosted on two Harris 1,200 CPUs plus ADI-100, contains about 100K SLOC in FORTRAN, and uses an Evans and Southerland CT-6 graphics system.

The value of a device such as CODES to an aerospace manufacturer is obvious. Although the cost is high, it is a bargain when compared with less effective options. The deficiencies of current development methods are real. Resolving them requires determination and commitment.

RECOMMENDATIONS

Although it is difficult to determine the technical content of a proposed paper from an abstract, more effort should be expended to avoid duplicating the very similar nature of many papers. It must be frustrating for a speaker scheduled late in the day to hear his/her key points and ideas expounded upon three or four times before he/she has a chance to speak.

It would also be helpful to symposium attendees if a hard copy of the transparencies used during the presentations were made available. These transparencies often have very different content from the published paper and synopsize the paper very well.

Since software development is a major part of the development of avionic systems, perhaps a special conference should be dedicated solely to software tools, techniques, and environments.

The symposium included extensive discussion of tools and programs to aid development activities. It would have been helpful if actual examples of tool outputs were shown.

Finally, although it is admittedly not a function of AGARD, some effort to influence or even effect standardization should be seriously considered by the executive committee.

LIST OF PAPERS AND AUTHORS

KEYNOTE ADDRESS
TO
53RD PANEL MEETING/SYMPOSIUM
OF THE
AVIONICS PANEL
NATO-AGARD
Las Vegas, Nevada
27 April 1987

Vice Admiral Glenwood Clark, USN
Commander, Space and Naval Warfare Systems Command

On behalf of the United States Government and the Department of Defense, I would like to welcome you to our country; Las Vegas, Nevada; and to the 53rd Panel Meeting/Symposium of the NATO-AGARD Avionics Panel. It is a pleasure for me to be your keynote speaker and to address such a distinguished gathering.

These meetings are an important element of the technology activities of NATO, giving us the opportunity to review the quality work that is under way within the allied countries of NATO. They offer a chance, from time to time, to establish and renew the personal contact between colleagues doing similar work. The close proximity of Nellis Air Force Base gives us the unique opportunity to see firsthand the tactical implications of our research and development work.

This group has set a very ambitious and vital task for itself when it explores how the system designer will build tomorrow's integrated avionics systems. As a senior military officer and a former program manager, I know we have successfully met the challenge of today's threats by developing and building quality systems. But, I am concerned about how we will meet the challenge of future threats, with the new technology and systems currently in development.

The papers to be presented here and the ensuing discussions will begin to chart a course through this new territory. It is easy to see that you are on target with the subjects like rapid prototyping, sensor fusion, Ada, multiaperture seekers, very high-speed integrated circuits, artificial intelligence, and generic architectures that are essential to the development of a new generation of systems. However, we must be sensitive to the implications of the total system.

I was pleased to accept an invitation to give this keynote address because in my current job as Commander, Space and Naval Warfare Systems Command (SPAWAR), I am fully involved with this issue. The total system, in its broadest, most generic sense, involves air, land, sea, and space-based sensors. It involves sea, land, and air forces working in concert as an integrated network or system. This is what I mean by the total system.

We face a potential enemy today that can field vastly superior number. Our strategy will be the use of superior technology combined with the ability to network our activities. We have the capability now, as never before possible in history, to integrate and focus our fighting forces into the ultimate total system.

For example, in a "war-at-sea" scenario, we expect some time in the near future to form a computer network of fleet air defense multipurpose fighter/attack aircraft, picket ships, and long-range surveillance aircraft which can share control of long-range missiles far beyond visual range.

The degree of information sharing between nodes of the network and the degree of controllability at various nodes will give the system tremendous flexibility and adaptability to meet anticipated increases in enemy capability.

That is the system concept in its broadest context. What you are discussing here will form a subset of this total concept. The technology, the tools, the methods, the advances that will be caused by you, ladies and gentlemen of AGARD, are vital to the overall effort. Keep your minds to new ideas and remember you are in the vanguard of this integrated approach to a total systems concept.

This total systems concept is uppermost in our minds within my command, where we are plowing new ground as we embark on designing the U.S. Navy's battle force as a system — more from the top down rather than the bottom up. We've never tried this before.

One might ask the questions: Why do we want to take on what is clearly a complex engineering management task? Why not keep doing what we've been doing, which has been reasonably successful? My answer is that we must change our way of doing business if we are to take full advantage of the technological explosion, and we must cope with our adversary's application of that same technology. Technology has

significantly shrunk the battle zone, whether it is sea, land, or air. This fact alone requires us to design a much more highly integrated battle force — one that is:

- Effective against potential threats.
- Assures us that we have functional redundancy, but only where we want it.

This requires a well system-engineered battle force.

We are already embarked on this total systems concept to which I have alluded, as a result of a major change in the way the Navy wants to design and acquire its battle forces of the future. This is no small task.

Heretofore, the application of a comprehensive system engineering approach in the U.S. Navy has been limited almost entirely to systems no larger than individual major weapon systems, e.g., FBM, AEGIS, and aircraft systems. We have never had the organization nor the dedication to approach the design of our total battle force through the application of systems engineering techniques and disciplines. We have now established that organization and signaled the intention to change our way of doing business.

I would like to talk about some of our early efforts during the past 16 months of work. Having first been tasked by the Secretary of the Navy almost two years ago to develop a process for system engineering the Navy, we were later tasked by the Vice Chief of Naval Operations to tackle battle force command and control as a first and critical opportunity.

We began the job of developing a BFC^2 architecture by doing that top-down and bottom-up analysis of Navy functions. We then aggregated these functions into loosely coupled major systems using a set of self-generated architectural principles.

We refer to these loosely coupled battle force warfare systems as (1) tactical command systems, (2) communication support systems, (3) warfare support systems, and (4) weapon systems, battle force command and control consisting of the first three. Let me give you a brief description of these battle force command and control "warfare systems" we have developed.

In our scheme, each of these systems would be specified by an operational requirement derived from a battle force top-level requirement. These systems would be designed as a system, budgeted as a system, and managed as a system. Now let me say a few words about each.

The tactical command system will be a repository of data. Its primary concern will be to the man/machine interface that supports Navy command authority. The principal objective is to enhance Navy command and control by providing our warfare commanders, at all levels, with an accurate and consistent tactical picture while minimizing the number of independent developments.

The communication support system can be envisioned as the "Ma Bell" system for the Navy. It treats all general-purpose data and voice communications, message standards, and communications processing elements as a system.

The idea is to pass from the other warfare systems the information that needs to be transferred to other Navy, Department of Defense, or allied units, and let the system process that information into the correct format for delivery.

Finally, there is the warfare support system. This system contains all wide-area and force-level surveillance systems, such as space sensors, and undersea surveillance systems, along with the sea- and shore-based systems that detect and correlate all contact data.

In the past, elements of this system have been treated quite separately as either shore-based or afloat-based systems without benefit of a top-down system architectural approach which would permit rigorous application of tradeoff analysis and system engineering disciplines in determining the optimum constituents of this system and defining/controlling interfaces between these constituents.

Our architecture calls for the engineering of these elements as one large system using system engineering disciplines that have proven so useful in designing major weapon systems.

Last November, the Vice Chief of Naval Operations approved this conceptual architecture, and we are now conducting a more detailed design definition and working to define a transition plan to this new system design for the three warfare systems. In addition, we have begun work to describe an architecture for each of the three major warfare areas: antiair warfare, antisubmarine warfare, and antisurface warfare.

This approach represents a new way of designing, budgeting, and managing the Navy's warfare systems.

There are many other activities that we are working on that bear on improving our ability to provide a well system-engineered battle force.

One of the most exciting things we are doing is an effort called the Fleet Initiatives Program. This effort is moving out quickly to couple more closely the Navy material establishment with our operating forces.

We have established a supporting organization for this effort, and we are working on new fleet-prioritized ideas which employ commercial computer resources to rapidly prototype tactical command systems. This effort provides an opportunity for us to learn the type of system that best supports our force commanders and allows us to write better specifications for development contracts. Where appropriate, it gives our fleet commanders an interim capability, and if the acquisition cycle continues to lengthen, it might be their only capability.

This program is also working on fleet priorities that cause changes to existing military systems for experimental work, such as third-party targeting.

Some of these prototypes involve new technology, like the artificial intelligence work the Defense Advanced Research Projects Agency is funding to build the fleet command center for battle management at CINCPACFLT. All of these are exciting efforts that will be continuing.

To continue our architecture work, we are building the databases of operational and system functions to translate top-level warfare requirements being written by offices within the Chief of Naval Operations into designs and engineering solutions for the appropriate weapon system.

We are also beginning to evaluate existing development option papers in the weapon systems to assure that proposed solutions meet the direction planned for the architecture. As an example, we are rethinking the electronic warfare control system to allow the necessary tight coupling between hard-kill and soft-kill weapons.

In the warfare system engineering work, we are coming to closure on the concept that we plan to use. This engineering work should provide more stability for our program managers. We will create warfare system performance specifications and warfare system control interface drawings as the mechanisms that will establish that stability between and among warfare systems.

I see these new thrusts in our Navy as exciting opportunities for the Navy to begin building a future that will sustain this country in its maritime strength and provide the President with the means to maintain our commitment to the free world.

I view these activities as the most challenging enterprise that the Navy has initiated in past decades. Now it is time for me to step aside and let this Panel get down to work.

Thank you for inviting me to be your speaker, and good luck in your deliberations.

TECHNOLOGY DEVELOPMENT PROGRAM FOR TWENTY-FIRST CENTURY AEROSPACE VEHICLES

William T. Suit and Douglas B. Price
NASA Langley Research Center
Spacecraft Control Branch
Hampton, VA 23665-5225
USA

INTRODUCTION

In 1985, a presidential directive initiated the National Space Transportation and Support Study which instructed NASA and the DOD to look at the national needs for large system architectures, avionics, and mission requirements to support the design of space transportation systems through the year 2010. The purpose of the joint NASA-DOD program resulting from the directive was to conduct studies to determine the technology development required for the design of the next generation of space transportation systems. These objectives are discussed in greater detail in reference 1. Much of the proposed research is mission motivated, and vehicle operations will be the focal point for this research. In this paper, we will primarily discuss NASA's input to the avionics technology development plan.

While control issues are critical in the design of propulsion systems and airframes, a major task is the development of operational simplicity leading to significantly lower costs. When discussing the next generation of space transportation systems words such as "autonomous," "adaptive," and "on-line" are used. If single vehicles are to be operated more efficiently or if multiple vehicles are to be supported by limited onboard crews and ground support staff, many operational functions will have to be handled by vehicle avionics systems.

Several examples will illustrate this point. Retargeting for new missions can currently require days for reprogramming, while the proposed vehicles will need the capability of retargeting in minutes. A current manned mission requires an extensive ground support network, while in the future, several manned vehicles could be operating at the same time with only limited ground or space station support available. The next generation of vehicles must, therefore, be autonomous and require only minimized outside support.

The research plan to develop the technology base required to meet the avionics needs of the 21st century space transportation systems was put together by technical representatives from NASA and DOD. It represents their best estimate of the research areas necessary to expand the current technology base required before actual vehicle design can begin. This program will serve as a basis for research requirements and the time necessary for their development. The plan developed has been divided into major categories. These are:

1. Guidance, Navigation and Control
2. Flight Systems Management
3. System Integration and Modelling
4. Communications
5. Man/Systems Interface

These categories will be further divided into subcategories with specific objectives. The extent to which this plan is implemented will determine how much of the suggested technology base will be available for the design of future space transportation systems.

This paper will assume full implementation and will discuss these categories and objectives with emphasis on the mission requirements they will satisfy. Since this is a long-term program that is just beginning, few results are available. The current studies, in many cases, are extensions of existing work and are general in nature, but are directed toward GN&C algorithms and flight systems that will result in an autonomous vehicle. A test program to demonstrate the concepts developed has been proposed, and this test program and its necessity will be discussed. A summary of the goals and potential impact of the technology plan will conclude the paper.

DISCUSSION OF RESEARCH CATEGORIES

Initially, the current state of the art must be established. The Space Shuttle will be considered representative of the current state of the art, and this paper will indicate improvements in the current systems required to meet the needs of follow-on vehicles. As the different categories are discussed, the results of the proposed research will be related to various proposed missions and experiments.

A discussion of the research categories follows:

1. Guidance, Navigation and Control (GN&C)

GN&C technologies will be developed to achieve on-demand launch/recovery, precision rendezvous and docking, and in-space operations without ground support. This effort includes developments required to achieve adaptive, all-weather, optimal, autonomous, fault-tolerant, onboard guidance, navigation and control. Critical tasks include the development of algorithms and the associated subsystems required to support autonomous GN&C. The associated subsystems include attitude reference systems, flight data sensors, actuators and supporting processors.

To accomplish the GN&C objectives, seven research elements have been identified. These are: 1. GN&C System Modeling, 2. Algorithm Development, 3. Sensors, 4. Actuators/Control Effectors, 5. Attitude Reference Systems, 6. Supporting Processors, and 7. Ground Test Simulations.

Mission needs require that various research elements or groups of elements be accomplished. New guidance, control and navigation algorithms must be developed to achieve the on-demand, autonomous, all-weather operation desired for the next generation of space vehicles. Improved instruments are required to support the algorithms developed, and these must be smart, self-checking, self-healing, and fault-tolerant to give a very reliable system. The concepts of redundancy and reliablity will have to be redefined in the new environment, and this redefinition will also be part of the proposed research. The object of the avionics systems for future vehicles is greater reliability with reduced maintenance.

The proposed research programs to accomplish the objectives outlined above are: 1. Algorithms for Autonomous GN&C, 2. Navigation System Requirements, 3. Subsystems Development, and 4. Vehicles Dynamics Modeling. Milestones for accomplishing various objectives in the research programs are shown as Table I.

TABLE I. Research Program Milestones For The Guidance, Navigation and Control Category.

YEAR	87	88	89	90	91	92	93	94	95	96
			1		2 3	4 5	6 7 8	9		

Milestones
1. Ascent guidance/control algorithms demonstrated
2. Autonomous navigation algorithms demonstrated
3. Rendezvous/docking guidance systems defined
4. Aero-assist Orbital Transfer Vehicle (AOTV) atmospheric guidance algorithms demonstrated
5. Air data sensor technology available
6. Attitude reference system technology ready
7. Autonomous automatic navigation system ready
8. Entry control algorithms ready
9. Unmanned GN&C system simulation completed

The research elements should be satisfied and the programs completed by 1996 so that final design of various aerospace vehicles can begin.

2. Flight Systems Management

Flight Systems Management technology advances are sought in three major areas: 1. Automated System Health Monitoring and Control, 2. Onboard Mission Planning and Retargeting, and 3. Flight Operations Management. The system health studies will permit advanced knowledge-based systems to perform self-test/diagnostic tasks, display risk evaluations and initiate appropriate reconfigurations or other corrections. Onboard mission planning/retargeting developments will allow fast reaction to changed mission requirements or parameters. Critical aspects center on the ability to develop and check-out adaptive, fast reactive software retargeting algorithms. Developed algorithms will be tested in laboratory simulations. Flight operations management techniques will allow real-time redirection or reconfiguration of subsystems. System tests will be conducted to demonstrate integrated technologies.

The research elements for the Flight Systems Management category are: 1. System Health Determination, 2. Onboard Mission Planning/Retargeting, and 3. Flight Operations Management. The health determination research element dictates the development of system checks to see if hardware/software combinations are working properly. Simulations will be designed to establish conditions to be met by GN&C algorithms for nominal missions and missions with changes. This onboard mission planning and retargeting will greatly reduce the ground support required during a mission. For manned missions, the information about vehicle health, retargeting or other automated mission operational functions will be displayed to the pilot in such a manner that he can react to this information and take any actions required.

The research programs proposed in the Flight Systems Management category are: 1. Autonomous Flight Systems Management, and 2. Systems Health Determination. These are general research programs, and the specific milestones showing when results can be expected are shown as Table II.

TABLE II. Research Program Milestones For The Flight Systems Management Category.

YEAR	87	88	89	90	91	92	93	94	95	96
		1	2 5	3	4 6				7	8

Milestones
1. Launch operations concepts defined and evaluated
2. Launch operations design requirements defined and evaluated
3. Launch operations algorithms developed and tested
4. Launch operations systems demonstration completed
5. Complete mission concepts defined and evaluated
6. Complete mission design requirements defined and evaluated
7. Complete mission algorithms developed and tested
8. Complete mission systems demonstration completed

3. Systems Integration and Modeling

Candidate avionics technology and subsystems architecture concepts will be explored to identify building blocks for custom configuration of autonomous, fault-tolerant avionics systems. Methodologies will be developed for evaluating various approaches to assure synergistic benefits, optimization and system compatibility. Critical tasks include system concepts definition and the derivation of analytical models for evaluating system performance and cost. A systems test bed is proposed to facilitate systpm level trade studies of conceptual design candidates.

The research elements for the Systems Integration and Modeling Category are: 1. Integrated System Concepts, 2. System Dynamics Analysis, 3. Systems Performance and Cost Modeling, and 4. Systems Test Beds. Design and testing programs will be required to meet the Systems Integration and Modeling objectives. These will be used to assess the impact of various design philosophies for systems ranging from computer architectures to combinations of structures and propulsion systems. The test beds developed will be used for on-ground certification of various design concepts. Advanced information-processing concepts will be required to support the programs developed. Some of the programs developed could be converted to operational flight aids.

The research programs initiated to satisfy the Systems Integration and Modeling category are: 1. Advanced Information System, 2. Aero-assist Flight Experiment Avionics Simulation, 3. Design Specifications, 4. Fault-tolerant Architecture, 5. System Performance Evaluation. The milestones showing the completion targets for different portions of the research are shown as Table III.

TABLE III. Research Program Milestones For The Systems Integration And Modeling Category.

YEAR	87	88	89	90	91	92	93	94	95	96
		1	2	3 6	4 7	5	8	9		10

Milestones
1. Prototype OTV/AOTV systems concepts defined
2. Prototype OTV/AOTV analytical models developed
3. Prototype OTV/AOTV systems concepts analyzed
4. Prototype OTV/AOTV systems test beds operational
5. Prototype OTV/AOTV systems trades completed
6. Multimission vehicle systems concepts defined
7. Multimission vehicle analytical models developed
8. Multimission vehicle systems concepts analyzed
9. Multimission vehicle systems test beds operational
10. Multimission vehicle systems trades completed

4. Communications

Jam-resistant communications technology will be developed for all mission phases, with emphasis on the on-orbit and reentry blackout phases. Current communication technology procedures will be reviewed to identify specific limitations and problems areas. Ongoing programs will be focused on space transportation needs. Critical concepts and disciplines include autonomous/secure communications, adaptive multifunction antenna systems, multiple beam formulations, and phased array semiconductor laser communications. Antenna system improvements will reduce reliance on ground station support; laser communication developments will be applicable to on-orbit operations, especially for data collection and reporting of self-monitoring system studies. Laboratory component testing will be conducted to validate the technology at the component level. The integrated system technology will be demonstrated in a flight test.

The research elements for the Communications category are: 1. Jam Resistance, 2. Reentry Blackout, 3. Antenna Design, 4. Coverage Bandwidth, 5. Data Transmission Rate/Reliability, and 6. Coding and Decoding. The communications system is to be designed to be operative throughout the operational range of the vehicle: from orbit, through in-atmosphere maneuvering, to landing. This is to be accomplished through antenna design and through the use of the laser techniques required to transmit large amounts of data.

The research programs for the Communications category are: 1. Focused Communications/Tracking, 2. Optical Wavelength Division Multiplexing, and 3. Real-time Image Processing Algorithms. The milestones giving the projected completion of portions of the research programs are shown as Table IV.

TABLE IV. Research Program Milestones For The Communications Category.

YEAR	87	88	89	90	91	92	93	94	95	96
						1	2	3		

Milestones
1. Component technology available
2. Flight test completed
3. System technology available

5. Man/System Interface

The human factors technology base will be extended to support development of an optimized man/machine interface through applications of automation and electronic display/control technologies. Methodologies for consolidating controls, integrating displays and improving crew station design will be established. This effort will identify requirements and opportunities for increasing performance with new control and display technologies, and determine high payoff items related to human factors, information management, and decision support. Emphasis will be placed on a systems approach to achieve intelligent man/system interfaces. Development of the ability to synthesize and evaluate candidate system concepts is an important element of this task. Element technologies will be validated by laboratory demonstrations.

The research elements for the Man/System Interface category are: 1. Crew Station Automation, 2. Control and Display Devices, 3. Intelligent Interfaces, 4. Human Factors, 5. Information Management, and 6. Decision Support. The Man/System Interface category is divided into two areas, remote manipulation and vehicle status. In either case, information must be passed to the crew so that the decision required to accomplish a mission can be made. Whether controlling a remote arm or retargeting a rendezvous maneuver, the displays that furnish the crew information must be simple but also complete. Properly designed displays can reduce crew workload and increase efficiency.

The research program established to meet the Man/System Interface objectives is as follows: 1. Control and Display Devices, 2. Man-machine Interface, and 3. Man-machine Interface Technologies. The accomplishment milestones for this research program are shown as Table V.

TABLE V. Research Program Milestones For Man/System Interface Category.

YEAR	87		88		89		90		91		92		93		94		95		96
				1			2		3			4		5		6		7	

Milestones
1. Requirements identified
2. High payoff areas established
3. Information management display developed
4. Preliminary integrated display developed
5. Integrated display technology available
6. Refined automated control system developed
7. Refined multimode pictorial displays developed

The preceding sections have outlined and given motivation for the research that will develop the technology necessary to design the next generation of space transportation systems. We will next discuss programs for validating and demonstrating the technology developments previously discussed. A figure from reference 1 (fig. 1) will help put the demonstration programs in perspective. In this paper, we are considering only the avionics part of the program elements shown on the figure. The projects proposed can be ground tests or flight experiments. In general, each demonstration incorporates a number of advanced technologies and may also utilize new operational concepts. These projects support the completion of appropriate milestones shown in Tables I through V.

REPRESENTATIVE DEMONSTRATION PROJECTS

The following is a discussion of five typical demonstration projects, which could be used to show that various technologies have been developed to the point where actual designs exploiting these technologies, can begin. The demonstration projects to be discussed are: 1. Optimized Guidance Algorithms, 2. Advanced Avionics Systems, 3. Autonomous Launch Capability, 4. Advanced Information Processing Systems, 5. Space Transportation System Communication Blackout.

1. Optimized Guidance Algorithms

The objective of this project is to provide validated guidance algorithms, indexed by mission objective and vehicle type, that permit real-time, onboard, and optimal flight path selection. This project will lead to a ground demonstration of Entry Vehicle landing and aerodynamic plane change capabilities in 1993. A flight demonstration of guidance algorithms for landing and aerodynamic plane changes, which would also support planetary aerocapture and landing technology readiness, is proposed for 1994. A demonstration in this time frame would support milestones 8 and 9 of Table I, milestone 7 of Table II and all milestones of Table III.

2. Advanced Avionics Systems

This project will validate modeling tools to enable in-flight subsystems/controls reconfiguration, demonstrate software for autonomous GN&C, demonstrate emerging photonics technology devices and processes for integrated optical control systems, and demonstrate advanced power distribution. The advanced power distribution system with fault-tolerant capability is to be demonstr ted in 1991. In conjunction with the Optimized Guidance Algorithms project, an autonomous guidance and navigation system design technology is to be ready in 1994. A photonic-based device system design will be demonstrated in 1994, and a complete avionics system will be demonstrated for an AOTV. This project will support milestones 5 through 9 of Table I, milestones 7 and 8 of Table II, milestone 9 of Table III, and milestone 6 of Table V.

3. Autonomous Launch Capability

This project will incorporate many of the results from the previous studies. Its objective is to develop the capability for prelaunch commit-to-flight determination based on onboard simulation, check-out, and loads prediction; and develop capability for onboard real-time retargeting, load relief, and abort decisions. The requirements will be defined for onboard simulation, check-out, loads prediction, environment measurements, and GN&C algorithms in 1991. This requirements definition relates directly to milestones 1 and 2 of Table I and milestones 1 through 4 of Table II. Instruments for onboard environment measurements will be available by 1994. The technology required for these instruments were shown as milestone 5 of Table I. A simulation for the prelaunch and launch phases of the demonstration will be available in 1994. This development is supported by milestone 9 of Table I and milestone 9 of Table III. A ground-based demonstration of an autonomous launch will be scheduled for 1995. The flight instrumentation and software will be developed by 1998 for a flight test on the Shuttle with parallel ground-based monitoring set for 2000. This flight test could be a partial fulfillment of milestone 8 of Table II and milestone 10 of Table III. The extent, to which this demonstration may verify the Man/System Interface research, has not been defined.

4. Advanced Information Processing Systems

The objective of this demonstration project is to provide validated technology for real-time fault tolerant and distributed control systems based on a sound theoretical foundation, mature reliability and performance tools and operating hardware based on fault-tolerant building blocks. Validation concepts for operating system hardware and software will be established in 1989. Also, a control architecture simulation for an OTV and the Aero-assist Flight Experiment will be developed by 1989. Design guidelines for hardware, software and validated performance and reliability tools will be established in 1991. Integrated real-time fault-tolerant distributed control technology and Advanced Transportation System control architecture technology will be ready by 1992. The Information Processing demonstration will impact the three preceding demonstration experiments.

5. Communication Blackout

The Communication Blackout demonstration has no direct link to the other research programs or demonstration projects, but its success will impact all missions. The objective of the demonstration is to develop antenna design and placement methodologies and techniques to minimize or alleviate communications blackout during high-speed, high-altitude atmospheric flight. Since the blackout is a function of the ionization layer structure about the vehicle, the development of computational fluid dynamics (CFD) codes to describe the ionization structure is required. Non-equilibrium CFD codes will be developed by 1989. CFD codes will be extended to estimate leeside free electron number, density, and ionization layer structure in 1990. Also, by 1990, techniques for alleviating blackout will be developed, including an experimental approach and instrumentation for their validation. CFD code validation for AOTV-type vehicles using Aero-assist Flight Experiment results will be available by 1993. The demonstration project relates directly to the Communications research program and supports all three milestones.

The successful completion of the demonstration projects should coincide with the maturing of the research programs to the point that the technology developed, at that time, can be considered ready to be incorporated into vehicle designs. Based on the technology available, decisions can be made as to which of the final designs are most practical (fig. 1).

CONCLUDING REMARKS

In this paper, a program to meet the avionics technology needs for the design of future space transportation systems has been outlined. A more general discussion of the proposed research program and its potential problem areas for both NASA and DOD is given in reference 1. This discussion gives greater detail for specific research programs. The program is designed to meet as many technology goals as possible by 1996 so that decisions can be made as to which vehicles are feasible and which should be constructed. The research program has been established based on mission profiles that are common to many vehicles. The time lines established for this research program are based on a full funding assumption and could slip or have to be limited if this assumption is not met. As of this time, priorities have not been set on the various programs listed.

The demonstration projects listed are only a sampling of the verification tests required to determine that the technology is ready to be used in a vehicle design process. An examination of the research program milestones shows that other demonstrations will be required to validate the results of all the research programs.

As pointed out, this paper describes a program that is just beginning. This is the first year in which significant funding of programs is expected. The ingenuity of many people will be tested to make the best use of these funds and to accomplish the program objectives. We look forward to an interesting 10 years.

REFERENCES

1. Walberg, Gerald D.; Gasperich, Lt. Col. Frank J. Jr.; and Scheyhing, Ernest R.: "National Space Transportation and Support Study/Technology Requirements and Plans," AIAA 86-1213-CP, June 1986.

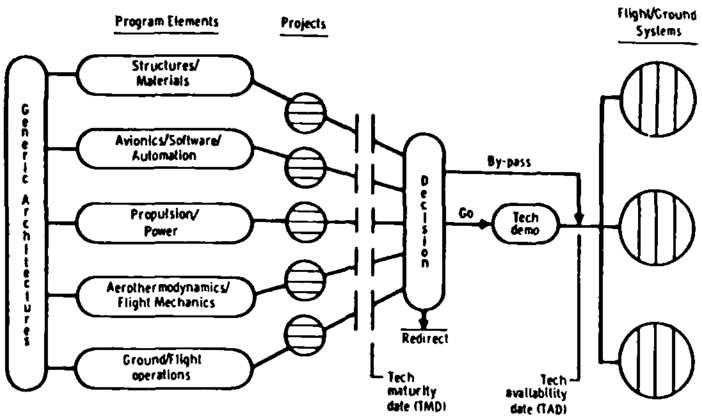


Figure 1. Demonstration Concept

DISCUSSION

C.Berggren, US

Have you quantified the real-time requirements for both subsystem reconfigurations and flight path selection?

Author's Reply

Studies are currently under way to address these questions. Results on the reconfiguration issue are expected in the 1988 time frame and on flight path selection in the 1989 time frame.

SYSTEMS FOR THE 21st CENTURY

Richard G. DeSipio
NAVAIRDEVCEN (Code 3013)
Warminster, PA 18974-5000
USA

SUMMARY

To a large degree the effectiveness and affordability of systems for the 21st Century will depend on the quality of "visioneering" which is applied today. Established ways and means of system definition, development and demonstration are rapidly giving way to a new set of criteria based upon system modularity, network interconnectivity and total system simulation. Avionics system engineering has become a sub-set of a total composite composed of avionics, platform and external interoperational environment. It is these three components which must be ordered as a total system.

In order to cope with what is on the technical horizon we must accommodate the transition from today's Black Box avionics to a future 21 Century modular partitioned architecture. An Avionic System Index will be proposed which allows for the definition of each function of the avionic system.

Also the results of two demonstrations conducted by the NAVAIRDEVCEN relative to the exploitation of on-board avionics built-in-test and diagnostics will be presented.

INTRODUCTION

A System is the contemplation of order achieved by direction of its implementation. Such a statement denotes *a transformation from the design of a system to the "Art of the System"* (1). Development and implementation of Systems for the 21st Century will demand the insight of the artist, the skill of a diplomat and a very high degree of management leadership. Avionics system engineering has become a sub-set of a total composite consisting of the avionics, the platform and the external interoperational warfare environment. Together they form the structure for fleet air operators to exercise their skills in order to meet mission objectives.

The primary cause for the scope of this totally is the operational need for coordination between elements of our forces and the time compression of emerging technologies. The challenge is for proper management and direction leading to the development, application and implementation of our avionics systems within the above context.

Advancing technology has caused the avionics engineer to work with an ever changing equation - *decrease equals increase.* The smaller devices become the more capability they provide. In addition, is the fact that today's design, development, and procurement practices are rapidly becoming outmoded to apply to tomorrow's systems. Simulations and computer aided design (CAD) have replaced the drawing board. Dedicated logic functions are being replaced with a computer on a chip. Human decisions are being replaced by machine decisions. Software no longer instructs a machine "How" to do, but rather "What" to do. This then is our starting point, a recognition of accelerating change and the resultant impact on the ways and means which we refer to as "business as usual". Additionally, we must recognize the interrelationships which exist between the avionics, the platform, and the warfare area environment. It is within this framework of reference that the systems of the 21st century must be implemented and operated. Together they form "The Art of the System". Just as a movie director must envision, coordinate and manage all actions on the "Set" *so too must the technical manager envision the operating* scenario(s), tactics and hardware for a given warfare area and set about to manage and direct the total composite. Further, it is precisely because of these conditions that we must set goals and objectives governing the application and implementation of emerging technologies if we are to achieve maximum effectiveness and at the same time be able to afford the promise of technology. We must have a management structure which provides guidance and direction in the form of an Avionics Doctrine or Policy. It is only by having a common set of rules and principles directed towards a common objective that we will be able to affort and provide the weapons system capabilities which will be required for the 21st Century.

DISCUSSION

Figure 1 presents the observation that the level of avionics system integration is directly proportional to the introduction and application of solid state electronics. The application of the transistor and *the microchip each created their own level of system integration.* Each allowed for an increased amount of compactness, required less power and increased the degree of overall system integration by allowing for the introduction of integrated control/display panels, embedded microprocessor equipments and overall system interconnectivity via a digital data bus. The next level of avionics integration is projected to "Step" in the 1989-1992 time frame. With the availability and application of VHSIC technology a new level of integration will be achieved. However, this will result in a "Giant Step" because of the increased allowability of related technologies. Smaller, faster, less power and greatly increased memory capacity has given rise to an era of software driven avionics such that the binary function has become "King". The introduction of VHSIC will open the door to numerous application of computer science in the form of data fusion, decision aids, expert systems and artificial intelligence (AI). The transition from the transistor to the *microchip resulted in the transformation* from single function black boxes to multi-function black boxes. The transition from the microchip to VHSIC technology will result in the transformation from the multi-function black box to a totally *integrated system* interconnected via a network of optical high speed data buses (HSDB), high bandwide matrix switches, and common backplane buses.

APPLICATION OF TECHNOLOGY STEP FUNCTION

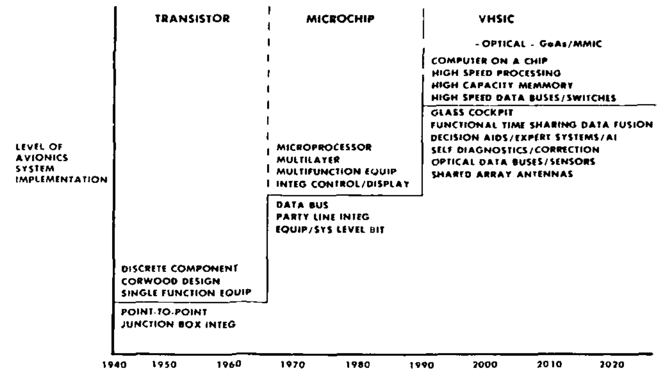


Figure 1

Figure 2 illustrates the reality of avionics implementation when compared to a specific period in time. Systems now in development such as, Integrated Comm. Nav. IFF Avionics (ICNIA) - Integrated Electronic Warfare Systems (INEWS) - Advanced Target Acquisition System (ATAS) - Integrated Inertial Sensor Assembly (IISA), will not be available at the same time. We don't have the luxury or reality of starting with a "clean sheet of paper". Equipment availability, present inventory and support logistics, and degrees of risk must be balanced and assessed in terms of specific platform schedules and funding profiles. In light of these realities we must apply a certain degree of vision by setting the stage for the introduction of new technologies. We must set up the framework for their introduction via an avionics architecture which will provide the allowability factor to incorporate technology insertion. We must therefore address a ways and means which offers us the opportunity to incorporate emerging technologies.

TIME LINE IMPLEMENTATION PROJECTIONS

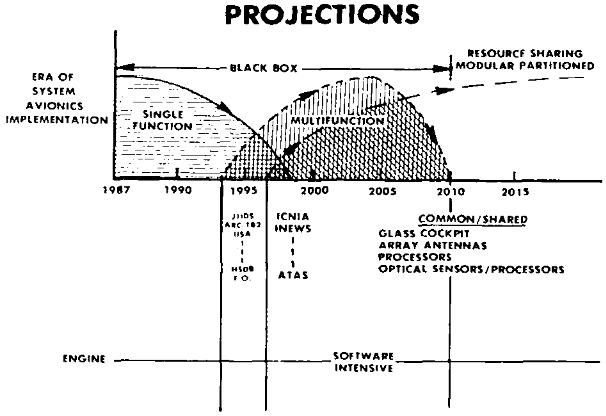


Figure 2

Figure 3 outlines the basic framework which must exist if we are to define a systems architecture governing external interoperability and intra-platform standardization. It is the existence of these interface networks which enables operators and machines to "talk" and function as a composite entity. Standardization is a key factor. Special purpose and/or unique systems are limited in application and costly to develop, implement, operate and maintain.

Within the external spectrum interoperability is made possible by standardization of messages, RF modulation and data link protocol. These parameters are implemented and controlled by various Tactical Data Interchange Link (TADIL) Standards, voice communications procedures and equipment specifications.

The standardization of an intra-platform network was begun with the introduction and implementation of the DOD MIL-STD-1553B and its NATO equivalent STANAG-3838 bus network. MIL-STD-1553B provided a significant step forward relative to avionics system integration. For tomorrow's systems we must continue with the development and implementation of internal aircraft network standards governing high speed data buses, wide bandwidth switch matrixes, and common backplane connectivity. These networks will serve as the framework for a future common Navy, military, avionics architecture.

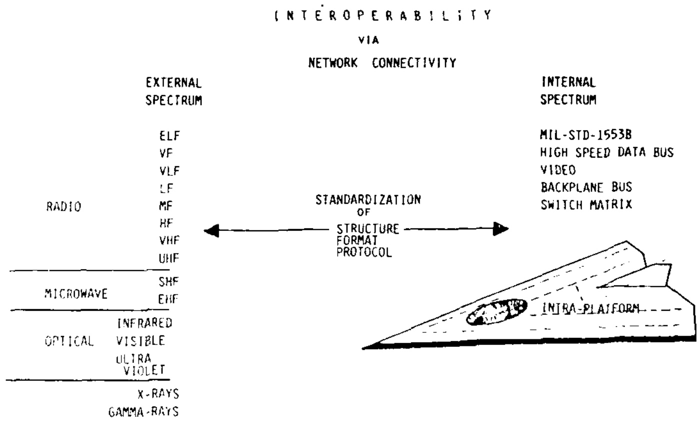


Figure 3

Figure 4 illustrates avionics architectural trends based on both government and industry projections. To be sure there are many other similar or varied versions both from within the government and from industry. However, they do reflect a common underlying thesis. That is, they represent a modular time shared partitioning of functions in order to group like functions into some type of managable architectural structure. It is to be noted that the key ingredient which allows for the interplay and construction of the avionics functions is the one of a common intra-platform network. This is the framework which supports the architecture.

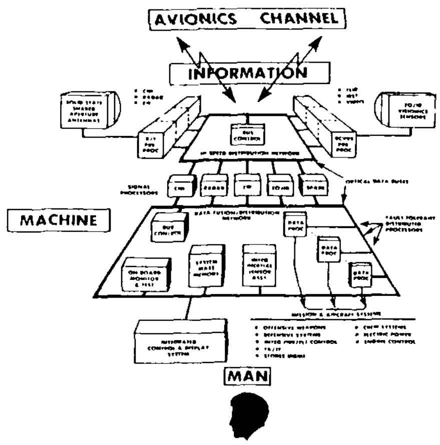


Figure 4

Figure 4 presents the concept of an Avionics Channel to replace the self contained dedicated channels implemented by today's black box systems. The advantage of this type of architecture is that it allows us to define functional areas of operation which can then be described by standard electrical/optical interface standards for networking throughout the aircraft. It is by so doing that we partition areas into functional domains and subsequently set up boundaries which can be defined in terms of function and interface, packaging and software standards. The intrastructure I have chosen is one of a System Index composed of Elements (E), Cells (C), and Zones (Z), Figure 5. The basic unit of the avionics system is the Element. Elements are implemented within a structure of Cells which in turn occupy a Zone. Such an arrangement is viewed as a means of defining, describing and locating any Element of a system similiar to having an avionics library of functions from which to choose and manipulate in order to construct and satisfy the requirements of a particular aircraft avionics system.

In practice the avionics system, Figure 5, is partitioned into three functional Zones. An Outer Zone containing all those elements required to accommodate the emitter-collector portion of a sensor system. The second zone is designated as a Gap Zone which provides the bridge between the modulated-demodulated signals and the baseband information processing which is turn will be accomodated within a third zone designated as the Inner Zone. The Zones in turn are divided into seven Cells each composed of specific Elements which are specified in terms of specific functions. As an example we might define and code an HF Link-II system as composed of Zo (E7C1+E3C2+E11C3+E9C4) + Zg(E6C5) + Zi(E14C6+E3C7). The procurement process would be for the development and procurement of each Element. The Element in turn is completely defined by an associated specification in terms of function to be performed, interface requirements, packaging configuration and support and operational software. As illustrated by Figure 5 each Element (Antenna - power amplifier - transmitter - receiver - preprocessor - signal processor - data processor - control - display) has a specific task within its Cell functioning either as a programmable reconfigurable device or as a dedicated single Element. In any case each Element is specific in its function and when combined with other required Elements will compose the entire system within an overall modular partitioned avionics architecture. One can further imagine the color coding of each physical Element associated with a designated zone for purposes of implementation and maintenance location.

SYSTEM INDEX

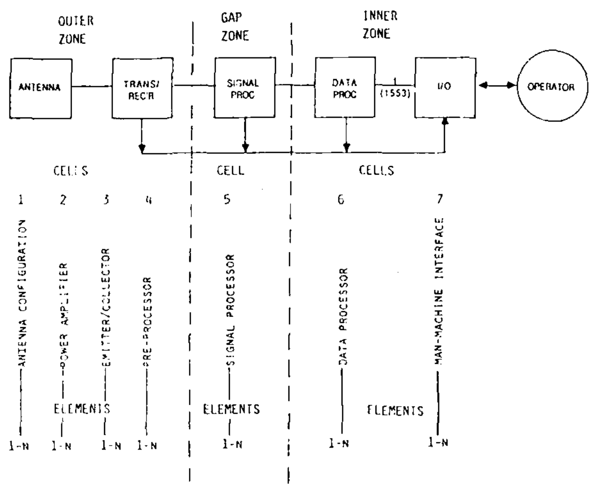


Figure 5

In order for the above concept to become a reality two assumptions are made. First, the military has as an objective the establishment of a common type of avionics architecture. Associated with this assumption is that a common avionics transitionary architecture is in effect along with a Doctrine or Policy which provides guidance and direction to allow us to evolve into the type of architecture of Figure 4. Secondly, there exists an approved standard Intra-Platform Interface Requirements Document (IPIRD). The IPIRD would govern implementation of networks, packaging, and software standards.

With these two assumptions in place and enforced we are in a position to take advantage of all that technology has to offer by having the means to exploit emerging technologies on an as come basis. Additionally, we can develop an avionics system data base, based on a common Avionics System Index, such that once established we can use automated tools of CAD and simulation to construct, exercise, and assess an avionics system well in advance of our commitment to invest in a complete system. We can validate how the system will operate and generate a number of what-if conditions to satisfy various operational and degraded modes of operation. Such a situation allows us to make our mistakes and achieve a lessons learned curve at our desk top and not at the flight line.

Also because a system index presupposes a rigid description and definition a specification can be generated governing each functional element placing the government in a position to generate procurement packages for a specific technology appropriate to that Element. Additionally by virture of its placement (Cell-Zone) in the avionics system such a scheme would provide a valuable means of describing functional domains which then set the stage for the introduction of expert systems and artificial intelligence operations. Bounding of a particular area of interest allows for the definition of functional responsibilities and interrelationships.

From the viewpoint of Expert Systems and/or AI each Element has in effect an encapsulated function such that we can concentrate on specific areas of applying AI technology. One very important dependency an AI Element will have within a system is that it must have access to a number of inputs as well as being able to disseminate its information to an intercom via speech synthesis and/or display systems or other processor for purposes of data exchange, consequently the need for standard interface networking. An area for the application of AI might be for the injection of natural language in terms of the pilot

questioning the status and performance of his avionics and aircraft systems. In this case an aircraft/avionics AI status Element would be defined in terms of what type of "input knowledge" would be acquired (from various on-board aircraft sensors and avionics equipment/system built-in-test), and what type of output information is required. By virtue of having a set of interface standards the designer of the AI Element would be in a position to access and formulate "source" data in terms of what information is wanted and when it is needed. Likewise once the AI processing algorithms are executed the information can be easily disseminated to a designated "sink". Another area of application might be for the identification and designation of an external platform detected by the aircraft's sensor suite. Additionally once identified the AI system could provide the complete weapons configuration and performance characteristics of the potential enemy platform to the pilot/crew. An AI Element programmed as an expert interpreter would be designed and implemented as a specific Element residing within the total system as for example Z1C6E23.

Earlier we mentioned the need for a transitional architecture which would allow us to evolve into the projected architecture of Figure 4. A transitional architecture would be implemented within upgrade and near-term (3-7 years) platforms. The transitional architecture will allow for the acceptance of emerging technologies by setting the stage for their insertion as they mature and become available. There are a number of "not needed right now" provisions, although at least one could be taken advantage of in today's contemporary data bus implemented aircraft. The MIL-STD-1553B data bus controller is configured to operate in two additional modes (inherent in 1553B protocol), the dynamic bus control and broadcast modes. Additionally, two data bus ports are incorporated. The first is for external access to the avionics system under conditions of "weight on wheels" for purposes of on-the-deck and remote station status monitoring/maintenance actions and avionics system initialization. This feature could be utilized for our contemporary data-bus configured aircraft. The second port is for the implementation of an internal 1553B to high-speed interface gateway for use at a future date. The next step is to seriously consider the MIL-STD-1773, the fiber-optic version of 1553B. We should take avantage of the benefits of a fiber-optic transmission media within planned upgrade and near-term future aircraft of the next 3-7 years. The transition to the 1773 could be accomplished by initially performing the electrical-to-optical conversion at the fiber-optic cable connection, not modifying each black-box electrical I/O. Such an approach would be cost prohibitive. Future equipments/elements would incorporate the optical 1773 as a basic I/O channel. Likewise, we should consider the installation of a fiber-optic high-speed transmission media to accommodate a future outer and inner zone high-speed data bus configuration. Once the stage is set for the insertion of future technology, we can begin to accept functional elements as they become available. This process is not unlike that employed for transformation from a totally hardwired system to a hybrid 1553A/B data bus system. Older equipments retained their hard-wired I/O and interfaced with 1553A/B compatible equipments via a gateway converter box, designated as a Communication Control Set, a Switching Logic Unit or some such nomenclature. In either case, the gateway converter coupler is necessary. As time goes on and a standard optical/electrical interface I/O is adapted for a 1553 and/or future outer and inner zone high-speed data-bus system, the gateway converter will no longer be required. A second step taken for the transition to a contemporary 1553 system was the incorporation of a growth factor within the bus controller. This allows for the addition of a number of black boxes to be coupled directly to the bus as manufacturers began to incorporate the 1553 I/O directly into a particular equipment. Today, this is the rule rather than the exception.

Of equal importance to the incorporation of emerging technologies is that we exploit the full potential of each technology. In order to illustrate what can be achieved within the bounds of today's technology and standard networks we shall cite two demonstrations performed by the Naval Air Development Center (NAVAIRDEVCEN). The first was a result of a concept proposed by the NAVAIRDEVCEN in 1972 to demonstrate the merits of implementing a common aircraft multiplex data bus system within a network of remote site service and support facilities Figure 6. In 1978 the NAVAIRDEVCEN awarded a contract for the development of a Multiplex Data Bus Test Set similiar to the concept equipment of Figure 6. The test set was validated in the NAVAIRDEVCEN LAMPS MARK III and Basic Avionics System Integration Concept (BASIC) laboratories. Subsequenlty the Air Force contracted for a similar but more comprehensive equipment in 1981 from LORAL Instrumentation, San Diego, CA. With the cooperation of LORAL and the Naval Air Test Center (NAVAIRTESTCEN), Patuxent River, MD a demonstration of the remote site service support system concept was conducted in June 1983 (2). An F-18A test aircraft stationed at the NAVAIRTESTCEN was connected to the LORAL Test Set SBA-100 equipment via the avionics 1553 internal data bus. A second LORAL SBA-100 was located at the NAVAIRDEVCEN, Warminster, PA. The two terminals interacted via RS232 as a remote site monitor and command message injection source. An around the world service support system is possible with such a system.

The second demonstration combined the use of emerging speech recognition and synthesis technology with the exploitation of the existing avionics system architecture of the Coast Guard HH-65A helicopter. An avionics System "Ask and Tell" (3) status information exchange was conducted via a device designated as an Aircraft Speech Interviewer (ASI). The ASI was developed under a contract to NAVAIRDEVCEN by Collins Avionics Division, Cedar Rapid, IA. The ASI was connected to the HH-65A 1553 data bus for injection of voice commands within the context of the data bus protocol and command status message field. Voice commands were converted to data bus words instructing selected equipments connected to the bus to execute their Built-In-Test (BIT) routines. Returning status words were received and converted to speech via a speech synthesis module. An interesting aspect of the "Ask and Tell" demonstration was the use of a (low skill) 13 year old operator to dramaticaly illustrate the advantages and utility which can be obtained within the context of our existing systems. This demonstration was conducted at Aerospatiale Helicopter Corporation, Grand Prairee, TX in September 1982.

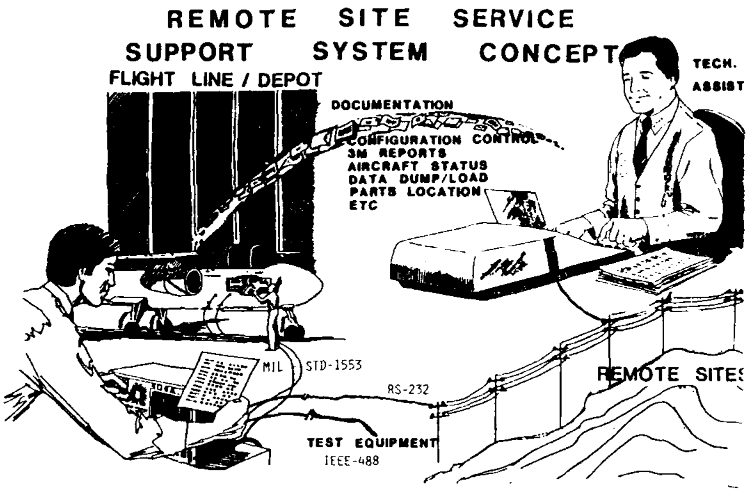


Figure 6

CONCLUSION

Emerging technologies together with the need for coordinated operations of our Naval and allied forces have created the need for standardized communications networks both within the aircraft avionics system and for purposes of warfare area interoperability..

Avionics system engineering has become a sub-set of a more broader warfare area engineering concept. Together the avionics, the platform and the external warfare area environment make up a total composite such that we must envision the future and apply "the art of a system" in order to bring each component into proper perspective and utility.

In order to facilitate the incorporation of emerging technologies on an evolutionary basis and at an affordable price we must have an approved and established policy in the form of an Intra-Platform Interface Requirement Document which articulates aircraft network connectivity, packaging, software and support standards.

Lastly we must set the stage for maturing technologies by incorporating a transitionary avionics architecture which will allow for the implementation of emerging technologies as well as the exploitation of the full potential of each technology in terms of expanding its utility.

References:

(1) Richard G. DeSipio, NAVAIRDEVCEN, Warminster, PA
The Art of Avionics System
Defense Science & Electronics, August 1986

(2) Philip J. Klass
Navy Test New Maintenance Procedure
Aviation Week & Space Technology, August 1983

(3) Richard G. DeSipio, NAVAIRDEVCEN   Eugene Fry, Collins Avionics Division
Avionics System Plays "Ask and Tell" with its Operator
Speech Technology, January/February 1983

DISCUSSION

P.Flahault, FR

(1) Does a MIL-STD-1553-2 exist?

(2) Notice 2 discusses communications means and recommends that the broadcast mode not be used. Please address this issue.

Author's Reply

(1) No, there are no plans to issue a MIL-STD-1553-2. The version issued in 1978 is the DOD (tri-service) version of record.

(2) Notice 2 does not "delete" the broadcast mode from the basic MIL-STD-1553B. Currently, there is much discussion within the tri-service/NATO/industry avionics community [e.g., Society of Automotive Engineers (SAE-AE9 Subcommittee)] on the merits of the broadcast mode. To date, to my knowledge, this mode is still inherent in the existing MIL-STD-1553B.

E.J.Manzie, CA

To clarify a point regarding MIL-STD-1553B, a Notice 2 to this standard exists. This notice addresses the broadcast mode and suggests it not be used.

Author's Reply

It is true that Notice 2 suggests the nonuse of the broadcast mode. However, the entire DOD (i.e., the Army and Navy) has not approved Notice 2; only the Air Force has suggested the nonuse of the broadcast mode.

ARCHITECTURE AND ROLE OF THE "SENSOR SUBSYSTEM" IN FUTURE AIRCRAFT WEAPON SYSTEMS

by

J.A.Salmon, C.J.C.Ravat and F.J.Lork
Thomson CSF/AVS
168 Bld. Gabriel Péri
92240 Malakoff
France

SUMMARY

To-day, in military aircraft systems, each external sensor (radar, ESM, electro-optical equipment, radiocommunications) is almost independent and reports directly to the central computer.

Taking into account first the evolution of operational context in which the aircraft is involved (increasingly varied and quantitative nature of the threat) and second the dramatic progress in processing capabilities (growth of several orders of magnitude in the same volume between 1980 and 1995) it appears necessary and possible to spread the "Intelligence" aspects of the weapon system in order to optimize the global cost-efficiency.

This distribution enables each specialist to concentrate his efforts, in order to take advantage of the ever increasing scope of knowledge associated with each discipline.

When considering an individual sensor, an expert system allows pseudo real time processing, taking into account simultaneously the characteristics of the environment and specific mission requirements. This is the high technology province of experts in physics and mathematics.

When considering the whole set of sensors, an expert system allows intelligent data fusion. This is once again the province of sensor technicians involving in addition operational aspects.

In the case of the whole aircraft weapon system, expert systems allow optimal control of the mission, taking into account the predetermined data and their updating in real time (tactical situation assessment, aircraft status, weapon status, mission planning man-machine interface, ...). This is within the scope of operational and mission managers, considering human capacities in environments of particular stress.

In this paper, the sensors subsystem is dealt with, its architecture is defined and its function in the aircraft weapon system is described.

KEYWORDS

Expert system, sensor fusion, external sensors, aircraft weapon system.

1. INTRODUCTION

To-day, in military aircraft systems, each external sensor (radar, ESM, electro-optical equipment, radiocommunications) is almost independent and reports directly to the central computer.

Taking into account first the evolution of the operational context in which the aircraft is involved (increasingly varied and quantitative nature of the threat) and second the dramatic progress in processing capabilities (growth of several orders of magnitude in the same volume between 1980 and 1995) it appears necessary and possible to spread the "Intelligence" aspects of the weapon system in order to optimize the global cost-efficiency.

When considering and individual sensor, an expert system allows pseudo real time processing, taking into account simultaneously the characteristics of the environment and specific mission requirements.

When considering the whole set of sensors, an expert system allows intelligent data fusion.

In the case of the whole aircraft weapon system, expert systems allows optimal control of the mission, taking into account the predetermined data and their updating in real time (tactical situation assessment, aircraft status, weapon status, mission planning, man-machine interface, ...).

In this paper, the sensors subsystem is dealt with, its architecture is defined and its function in the aircraft weapon system is described.

2. PREDICTABLE GLOBAL ARCHITECTURE

Many papers have already presented various architecture which could be proposed for future military aircraft.

Among the most significant, we have retained the architecture proposed in the DARPA project, the PILOT'S ASSOCIATE [1].

The general diagram of the PILOT'S ASSOCIATE is shown on figure 1.

In this diagram, the "black-box" named "sensor fusion" is a key element of the operational situation assessment in real time. As a matter of fact, it is the relevance of the information issued from this "black-box" which provides the pilot with the necessary decision factors to ensure the success of his mission and his own survivability. Ronald M. Yannone [2] discusses, in his article, the role of this "sensor subsystem" and its close relationship with the tactical situation assessment subsystem.

This type of architecture seems to us to fit in very well within the capabilities of the different "actors" involved in the design an development of an overall aircraft weapon system.

The various external sensors considered in future military aircraft (radar, ESM, electro-optical equipment, radiocommunications, ...) have each their own specific characteristics and the men in charge of these equipments must be very highly specialised in their respective fields.

At the other end, the man in charge of the whole aircraft weapon system, must be a specialist in operational and mission manager problems.

To prevent overload and therefore inefficiency, the pilot can only take into account synthetic information, cleared of their fluctuations, their uncertainties and sometimes their contradictions.

Due to the great variety of information issued by the sensors, and their possible complementarity, it appears necessary to achieve a "smart" fusion, in order to improve them and to solve the conflicts which might occur.

The responsibility for this "smart" fusion can only be assumed in an efficient way by the men who have very full technical knowledge of the different sensors. They must also have a fair knowledge of operational problems in the avionics field.

It's only with this combined proficiency that they will be able to solve, for instance, the identification conflict in which an "error" of a sensor can be due to a kind of counter-measure from the enemy.

3. DESCRIPTION OF THE SENSOR SUBSYSTEM

At the date we are considering, i.e. the years 2000^{+}, the external sensors equipping a sophisticated military aircraft will be the following ones :

- a radar for detection in the frontal zone (at least a cone of 120°) with an IFF interrogator,

- a passive detector of radio and radar signals (ESM) covering any direction, (4π steradians)

- a passive and active (Lidar) electro-optical system, covering any direction,

- a multifunction communication system such as the SINTAC (French equivalent of JTIDS).

These "sensors" will include very sophisticated data processing, including expert systems, in order on one hand to optimize their own functioning in the real environment, taking into account the initial constraints (type of mission, discretion, ...) *and on the other hand to perform a certain level of* identification on the detected targets (for instance, Doppler lines of aircraft engines "seen" by the radar, identification of a given terrain following radar by the E.S.M., significant loads of missiles under an aircraft "seen" by the E.O. system, ...).

A good example of this identification function is given by Thierry SCHANG in [3].

The sensors will provide two types of information :

- numerical data, with a quality factor (σ),
- symbolic data, with a confidence factor (C)

Table 2 summarize the information provided by each sensor.

The essential characteristics which appear when examining those data are the following ones :

- They are not simultaneous (different rate) nor are they all present (difference in detection range, for instance : ESM 200 km, Radar 100 km, E.O. 20 km in a given situation, ...).
- Their avalability depends on the type of sensor (ESM generally provides neither radial velocity nor range), and on the jamming environment.
- They do not have the same accuracy.
- They are not all reliable (use of decoy by enemy).

Therefore the interest of a "smart" fusion is evident in order to improve the accuracy and the reliability of the information (especially in presence of counter-measures).

The block diagram of the fusion system which could be realized is given in figure 3.

The numerical data (NUM) from each sensor will generally be provided to the system only after filtering or tracking in order to increase their quality.

The symbolic data (SYMB) delivered by their internal expert systems will be refreshed in quasi real time.

The overall data coming from each sensor enter a processor allowing their temporal and spatial alignment (1).

The numerical data are then sent to a proximity estimator (2) in order to obtain an initial spatial correlation.

The so-defined "sets" are then sent to an expert system (3) which will take into account symbolic data to effect an evaluation of coherence. The result could be a rejection of the proposed correlations and possibly new proposals.

The dialogue will converge on the generation of identified "tracks" (4) with a certain level of confidence (C). These tracks will be continuously maintained (5) in order to :

- to confirm and improve,
- cancell or reject (too old or level of confidence too low),
- propose new tracks to the processor (4).

For each detected target, the "track" to be carried out will consist of the following information (with their quality or confidence factor) :

- Status vector X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$.
- Range
- Radial velocity and time-to-go before direct threat (according to criteria)
- "Smart" identification (i.e. MIRAGE III E + weapons A, B, C, ...).
- Threat data, for example :
  . detection probability by "his" radar,
  . probability of triggering "his" weapons (lock-on range, shooting range).

These data are then sent to the tactical situation assessment Expert system, in order to enter the whole "Pilot's Associate".

4. CONCLUSION

Highly decentralized and adaptable avionics systems are the only way to effectively counter the ever increasing complexity of weapon system :

- decentralization facilitates the drawing up of module specifications and therefore their test and validation,

- flexibility, made possible by the Artificial Intelligence approach, allows updating of complex systems with minimum maintenance of embedded software.

A lot of work remains to be done to build a complete and efficient electronic pilot's assistant. In addition close cooperation between the different "Experts" of the overall system is necessary :

- Experts on the different sensor aspects who have to provide pertinent information from the outside world.

- Expert on sensor fusion, who must take into account sensors capabilities on one hand, and operational problems on the other hand.

- Expert on overall aircraft aspects who must take into account all the mission constraints and the human pilot's capabilities.

The winner of the aggressive competition in the military avionic market will certainly be the country that will achieve both a friendly and interactive cooperation between the different companies involved in the design of the future systems.

5. REFERENCES

(1) Kenneth J. STEIN - DARPA Stressing Development of Pilot's Associate System AW & ST April 22, 1985 p. 69-74.

(2) Ronald M. Yannone. The role of Expert System in the Advanced Tactical Fighter of the 1990's 0547-3578/85 - 1985 IEEE.

(3) Thierry SCHANG - Threat Identification : an Artificial Intelligence Approach. AGARD AVP 52th Symposium.

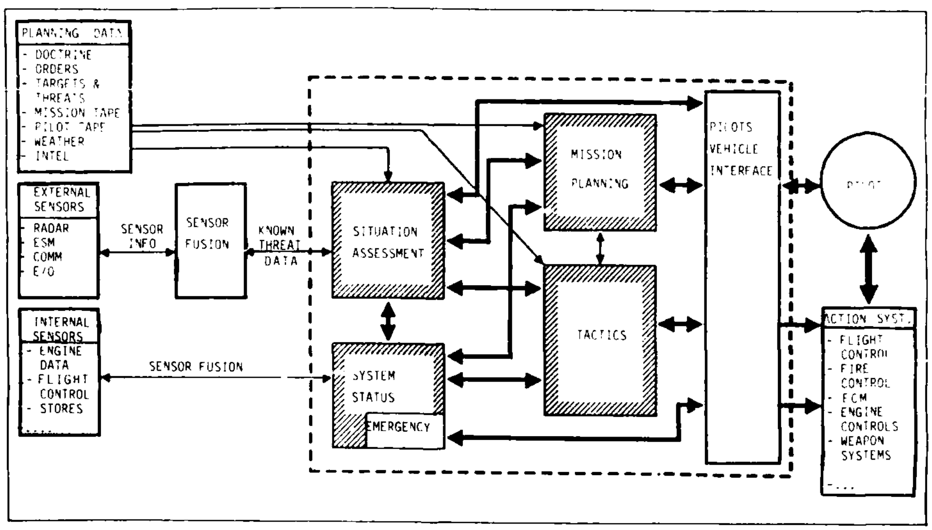


FIG. 1 - PILOT'S ASSOCIATE

SENSORS	NUMERICAL DATA	SYMBOLIC DATA
RADAR	Range, θ_s, θ_g, v_r, (γ_r) (X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$)	Radar cross section, fluctuations, "profile", engine lines.
IFF	Range θ_s, θ_g, (for friend only)	Friend identification
ESM	θ_s, θ_g (Range possible)	Radiating equipment type (radar, radio, IFF, remote control)
E.O.	Range θ_s, θ_g, v_r	Visual type identification Optical signature
RADIOCOMMUNICATIONS	X, Y, Z $\dot{X}$, $\dot{Y}$, $\dot{Z}$	Full or partial identification

FIG 2 - DATA ISSUED BY SENSORS

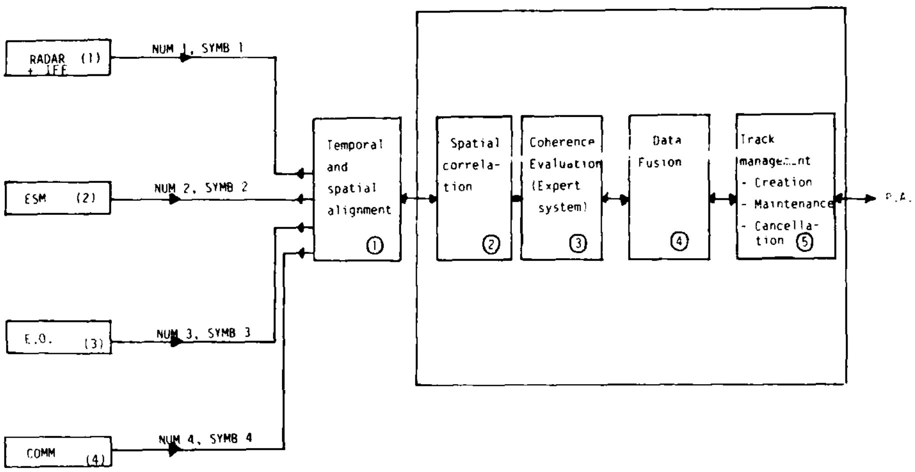


FIG 3 - GENERAL DIAGRAM OF SENSOR SYSTEM

DISCUSSION

W.R.Fried, US

(1) Will the French data system you mentioned be compatible with the US JTIDS (e.g., modulation and message format)?

(2) Does the planned processing include automatic correlation of ESM-derived (INTEL) target tracks and radar-derived target tracks?

Author's Reply

(1) Yes, the Systeme d'Identification, de Navigation, de Controle de Trafic, d'Anticollision, et de Communication (SINTAC) is fully compatible and interoperable with JTIDS TDMA and has the same modulation and message formats (i.e., Link 16/TADIL J).

(2) Yes, the system includes ESM tracks and radar tracks correlation.

E.Cambise, IT

(1) What is the time frame of the implementation of the proposed architecture?

(2) What are the hardware requirements for processing power and memory to support the implementation?

Author's Reply

As mentioned, we are in a preliminary study phase. It implies that:

(1) The architecture will not be implemented on the next-generation aircraft, but on the next 2000 year generation.

(2) Processing power and memory requirements are not yet defined, but we estimate that they will be at a very high level, achievable only with the most advanced VHSIC products.

R.Guiot, FR

Y a-t-il une applicabilite pour les avions de la période 1991–1995?

Réponse d'Auteur

Il y a en etude, sans beaucoup d'intelligence artificielle, un couplage radar-infrarouge.

RAPID PROTOTYPING OF COMPLEX AVIONIC SYSTEM ARCHITECTURES

by

L. Berardi, M. Giorgi, W. Mellano, A. Valente, E. Zucco

Aeritalia - Gruppo Sistemi Avionici ed Equipaggiamenti
10072 Caselle (Torino) - Italy

Summary

This paper describes a design tool called ECATE (Expert Consultant for Avionics System Transformation Exploitation) developed by the Avionics System and Equipment Group of Aeritalia.
ECATE, rapidly prototyping different alternatives, helps the designer in establishing the information flow architecture of the avionics system, that is the organization of the internal data handling. The tool provides the user with an interface to assist him in describing the avionics from the point of view of the data handling, and presents the results in a suitable format; it performs consistency checks and advices the user on possible architectural problems by means of the expert system techniques.
The paper contains also some indications on the development environment of the tool and how it works in a consulting session. Some examples give an idea of the result that can be obtained.
Conclusion is that not only the tool is valuable for the information flow architecture design but also it shows that the use of the knowledge engineering and the Artificial Intelligence techniques can be effective to meet the problems arising when complex systems, not only avionics, are involved.

1. Introduction

A state-of-the-art avionics system shall be fully integrated with the other systems of the aircraft and shall take full advantage from the features of the microelectronics, to provide the crew with the highest mission success probability.
It means to find a real implementation for concept like distributed processing, sensor data fusion, adaptive reconfiguration, expert pilot assistant, synthetic world displaying, now made possible by the advancemement of the technology, specially the data processing and transmission.
But in such a system the performance increase is not simply due to the higher performances of the microcircuits, on the contrary it derives primarly by an increase of the overall system complexity.
In fact the "black box", a large unit with well defined interface and function allocation, is no more the basis for the advanced system design but is being substituted by lower scale units, which changing combinations provides the best adptation of the system to a changing environment.
It is difficult to establich a metric for the system complexity (see for example ref. 1), however it could be said that it is reflected by the amount of memory used for operational software storage, which today is increasing to a rate at least an order of magnitude higher than the number of other microcircuits in an avionics system.

The increasing complexity, while can allow for dramatic improvements in terms of reduced pilot workload and mission success probability, has also some important drawback.
It is evident that a complexity which is mainly software implies a design, development and testing process and a management of it much more difficult than in a conventional system.
Therefore some change to the way of thinking, the tools and the organization of the industry are necessary.
It is our opinion that the tools and techniques of the Artificial Intelligence can be effectively applied not only in the system itself, but also to manage its complexity for the people who shall design it.
Scope of this paper is to illustrate an application of the rapid prototyping and expert system techniques to the design of the data flow organization within a complex avionics system.

2. The problem of the system architecture

The problem concerns the establishing of a correct data flow architecture. There are several interpretation of the term "architecture" in the avionic system design; it can be applied to the physical structure, the topology, the software organization and so on. All these are aspects of the same characteristic, the way in which the system components are organized and work together in order to create a system.
The architectural aspect chosen for the application described in this paper is the information handling within the avionics, i.e. the characteristics of the data flow and processing among the various system components, considered from the point of view of the information treatment.
Therefore the following definition of architecture will be usued in the following paragraphs:

Definition

System architecture is the organization of the information generation, distribution, processing and utilization within the boundaries of the avionics system.

A pictorial view of the above definition is given in fig. 2-1.
The boundaries of the avionics system are intended to define the meaning of generation and utilization of the information.
In other words if the boundary identifies the world outside the aircraft all information coming from it corresponds to a generation of information for the avionics system; on the other side the data are utilized when they are provided to the crew via a display or to the external world via an antenna.
Such an architecture is relatively easy to describe by means of few building blocks with a limited number of peculiar characteristics; but a correct design of it has relevant influence on the overall performance of the system, because it is usually established in the very early stages of the design and it is difficut to be drastically changed during the development process.

Therefore it is clear that a serious error in the data flow architecture design impairs the achievement of the design objectives in terms of time, cost and performance.
For that reason the architecture of the avionics system is usually designed by highly experienced people with support of the operations research tools (see ref. 2): nevertheless the work of these people is difficult to quantify and to describe analitically, being often result of empirical knowledge and heuristics.

When the complexity grows the difficulty of the design task dramatically increases to a level that the problem shall be partitioned into simpler problems, loosing part of the efficiency achieved by an overall view.
An alternative that can help to still consider the problem from a global point of view keeping to a reasonsable level the complexity managed by the designer, is to take advantage by the Artificial Intelligence tools and techniques, prepared to describe and process complex situations with an heuristic approach to the problem, i.e. rapid prototyping and expert systems.
Rapid prototyping of a complex architecture helps to easily evaluate many alternatives while an expert system directs the search for the best design. A dedicated tool combining together the two techniques can organize and manage the overall complexity of an architecture, requiring from the operator higher level decisions only.

A tool like that sketched above, described in the following paragraphs and developed in our laboratory, can be of effective use for the purpose and can demonstrate the advantage of the Artificial Intelligence approach in the design of complex avionics systems.

3. Theoretical considerations

3.1. System description

The building blocks that shall be used for the construction of an object oriented data flow architecture have characteristics that describe mainly their attitude with respect to the information handling.
Four types of objects represent the building blocks.

1. **Generators,** the sensors of the system, the controls available to the crew and the interface to other systems.

2. **Processors,** signal processors, mainly associated with sensors and displays and data processors to elaborate information at an higher level.

3. **Utilizers,** displays for the crew, interfaces to other systems, emitters or weapon which stimulate the external world.

4. **Channels** transmission means that link together all above objects when not directly interfaced (aggregation of objects).

The table 3-1 lists an example of the typical characteristics associated to the objects.
It shall be pointed out that the characteristics may vary in relation with some peculiarities of the described system.
The processors and the channels are possibly multiport devices, while equipment like a monostatic radar may be described by a signal processor, a generator and an utilizer, that is an aggregation of objects.
Although not directly related to a technological solution, the objects that form a system architecture from the point of view of the information handling, shall nevertheless take into account the state-of-the-art to avoid a design perfect but not feasible.

The building blocks shall be combined to form the information handling architecture corresponding to the functional architecture to model.
The architecture is characterized by some features, system descriptors which are listed in table 3-2.
Some descriptors need explanation on its definition, while the calculation methods are embedded into the tools and will be described in para. 3.

Risk The development risk take into account how much each object is close to its technological limit and how the the combination of objects influence the development.

Integration level It takes into account how good is the processing within the system. An higher integration level is a merit.

Growth Capability Represents the dual of the resource utilization of processors and channels.

It shall be noted that the descriptors can be computed also for a limited portion of the system, a subsystem.

3.2. Rapid prototyping and expert system design

A rapid prototyping tool shall assist the user to convert from the functional/performance requirements to a description that uses the object and connections illustrated in para. 3.1.
But an easy means to prototype many alternate design solutions is not sufficient because the knowledge behind the architectural design is not totally convieed by analytical descriptors.

Therefore an expert system, a tool that allow to acquire, use, modify and make available a type of knowledge which is complex, difficult to transfer, empiric, incomplete and heritage of a limited number of people is the most appropriate supplement for the rapid prototyping tool.
The expert system shall direct the search for a better architecture and provide advice on solution that may also not have different descriptor values but are known to guarantee an higher confidence of success.
The operational flow of an architectural design carried out by means of a rapid prototyping expert system is skectched in fig. 3-1.

4. The tool, ECATE

4.1. The environment

The tool, foreseen by para. 3.2 and called ECATE (Expert Consultant for Avionics system Transformation Exploitation), has been developed by means of KEE (Knowledge Engineering Environment, TM by Intellicorp), running on a dedicated LISP workstation (EXPLORER, TM by Texas Instruments).
KEE is a development environment prepared for Expert System construction, it could be considered an hybrid tool built on a range of state-of-the-Art Artificial Intelligence techniques utilized to combine different type of knowledge.
The knowledge is organized in frame/units associated to which are their peculiar characteristics, that structure is particularly suitable for the description of our problem because it implements a programming oriented to object linked by relations.
By means of KEE it has been implemented the following:

1. The user interface

2. The collection of objects and relations that represent the system

3. The algorithms and procedures for the descriptors computation

4. The expert knowledge

5. The knowledge handling structure.

The knowledge about the system is acquired via a graphic interface and processed by the inference mechanisms embedded in the KEE environment, according to the set of rules describing the expert knowledge.

4.2. The structure

The structure of the tool can be described by means of the block diagram shown in fig. 4-1.
Hereafter follows a brief description of the main components of the structure.

User Interface The user interface assists the user to represent his system in accordance with convention of the formal description and is formed by:

a) graphic utilities using icons, representing the objects, with associated menus for describing their characteristics.

b) indicators of the system descriptors of the terminated system.

c) menu sensitive "pushbuttons", i.e. means to activate a "method" (see below).

Methods The methods are procedures codified in LISP to execute algorithms, object interaction and reasoning/control stategies (see table 4-1 for example of methods).

Permanent data base It contains the description of the four types of objects and their classes . It contains, moreover the expert system rule base, unmodifiable by the user.

Working area It is formed by the units, which characteristics, called "slots", describe all information about the system under development.

Inference structure This structure is formed by an inference engine operating on the rules.

The structure and the development environment allow for the maximum flexibility; to change the object and system descriptors, inference rules or methods is extremely easy for the expert system designer. That feature is of capital importance and is used currently because the tool shall evolve with the knowledge available.

4.3. Consulting ECATE

The steps of a consulting session are summarized in fig. 4-2 and briefly explained hereafter.

Configuration Insertion The user, with assistance of the tool graphic facilities inserts the configuration of objects, aggregations and relations he wants to prototype (see for example fig. 4-3).

Compatibility verification The tool verifies after each input its compatibility with the objects related to it.

Overall Compatibility When the configuration is complete the activation of the "terminated system" push-button starts a verification of the overall architecture compatibility.

The results of the step above can be:

1. *Request for more information (for example some relation is lacking or some data are not available)*

2. Display of incompatibility warnings at system level (for example a multipoint channel of insufficient capacity).

3. Satisfactory compatibility

Descriptors computation When the system compatibility is not violated a method is activated to compute all system descriptors for which sufficient information is available.

Result display The results of the previous step can be displayed (see fig. 4-4) in either the numerical or hystogram format.

Optimization ECATE, by means of rules activated in forward chaining inference presents to the user advices on possible architecture problems and suggested changes involving objects, aggregations, subsets or the overall system.

Assistance request The use can ask for assistance in optimizing the architecture, giving, if he wish, instruction on the direction of the optimization.

Configuration change In case the user wants to follow one or more of the devices he can, by means of a pushbutton, modify the configuration and restart the consulting.

Explanation The user can ask at any time information about the facts that have activated the rules and generated the advices.

The tool accepts at any step not only numerical values in response to its queries but also generic indications, like high, low etc.
The consulting session can be terminated at any time and the results saved in the library.

4.4. Validation

The validation of a tool like ECATE shall answer to two kind of questions:

a) Is the tool conform to its specification?
b) Is the tool suitable for its purpose?

For the first check ECATE has been submitted for evaluation to the experts who concurred in its development and to foreign experts, exercizing it by means of test cases.
The second check is much more difficult to perform, because it is supposed to require a demonstration of the development of different architectures, accepted or rejected by ECATE.

The validation is still in progress, but the results available, related to the first check, show a good agreement with the predictions.
Nevertheless it shall be pointed out that the high flexibility allowed by the development environment and the tool structure stimulate a continuous refinement to adapt ECATE to new situations or to increase its knowledge. It is in fact current practice to introduce new object descriptors, computing methods or inference rules.
Therefore also the validation is continuing to follow the evaluation of ECATE and will not give final results until the refinement work is completed.

5. Examples

5.1. A simple system

In this paragraph it is shown an example of a simple architecture to illustrate how the system works.
Fig. 5-1 shows how the tool allows to represent the sketch of the system prepared by the designer, while fig. 5-2 shows how the entire system architecture looks like and comprise some advices given by the expert system.

5.2. A more complex system

A state-of-the-art system with complex architecture is shown in fig. 5-3, already in the formalized description of ECATE.

5.3 A Future avionics system

At the moment we believe that the knowledge available on future avionics system architecture, (see ref. 3), is not sufficient to effectively use ECATE.
Reason is mainly because, although enough data on sensors and processors can be found, the knowledge lacks in the display and control area and specially on standardized multipoint channels, switch, backplane and high speed data bus. Insufficient is also the knowledge of the rules that regulates the overall system functioning.
Nevertheless data are gathered and trials are performed with reference to experimental data to allow the specific ECATE knowledge to improve.

6. Conclusions

The scope of ECATE, an architectural design tool conceived and realized by Aeritalia, Avionic Systems and Equipment Group, in twofold.
First, it shall provide valuable means to design the information handling architecture of an avionics system, that is a rapid prototyping expert system, which make available a knowledge difficult to acquire and transfer and often heuristic.
Second, it shall demonstrate the effectiveness of the Artificial Intelligenc tools and techniques in managing complex design problems.
Although ECATE is still in its first stages of evolution, mainly about the inference rules and the user interface, it already shows a remarkably good achievement of the first objective above.
But the best result is obtained in demonstrating the second objective, because ECATE shows an excellent flexibility in the continuous refinement of all its parts and a rapid response to the changes introduced by the user.

The latter feature, made possible by the Knowledge Engineering Environment of development, is greatly valuable in the architectural design, because it makes available to the designer comparison between different solutions considered from different points of view and in overall.
It is our belief that the tools and techniques of the Artificial Intelligence can be applied also to other the other phases of the design, development, testing and to the management of avionics, and non avionics, system when complex problem are implied.
The benefit given by the intrinsic flexibility of the powerful knowleddge management techniques, greatly surpasses the initial cost of training people and acquiring tools, and results in a better more effective product.

Acknowledgements

Aeritalia, Gruppo Sistemi Avionici ed Equipaggiamenti, wish to thank UNISYS, Italian branch, for the support they provided in the use of KEE (TM) and Explorer (TM).
Responsibility of what stated in this paper is nevertheless only of Aeritalia, Gruppo Sistemi Avionici ed Equipaggiamenti.

References

1. System Complexity : Its Conception and Measurement in the Design of Engineering Systems - Devendra Sahal. IEEE Transactions on Systems, Man, and Cybernetics, June 1976.

2. *Critical Factors and Operational Research in tactical fighter avionics system development* - L. Berardi Conference Proceeding no. 343. 45th AVP Symposium Paper No. 29.

3. New technology impacts on future avionics architectures. - R.S. Mejzak Naval Air Development *Center, PA, USA. Paper No. 10 of the 51th AVP Symposium.*

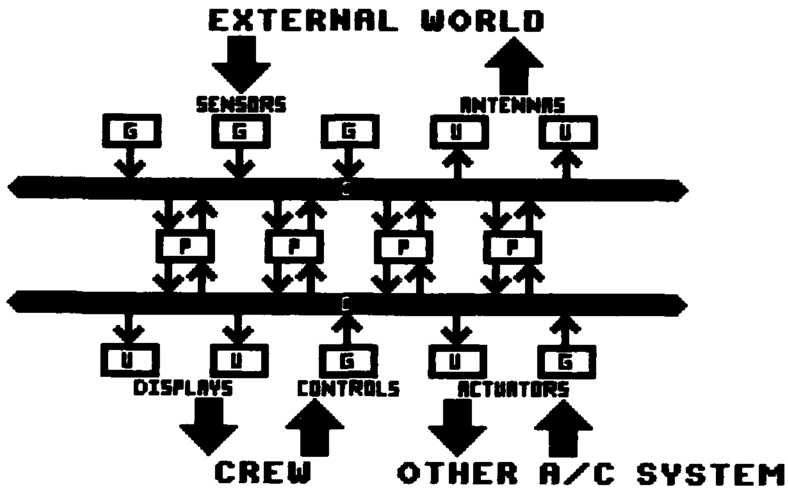


Fig. 2-1

GENERATORS Slots
member:
CONNECTED-TO-OBJECT
COST
GENERATOR-TYPE (local)
MINIMUM-LATENCY
MTBF
OUTPUT-INFORMATION-FLOW (local)
OVERHEAD
REDUNDANCY
RISK-FACTOR
SIGNAL-CRITICITY
SIGNAL-TYPE

CHANNELS Slots
member:
CHANNEL-TYPE (local)
CONNECTED-TO-OBJECT
COST
INFORMATION-FLOW-CAPACITY (local)
INPUT-INFORMATION-FLOW (local)
MAX-NUMBER-OF-CONNECTIONS (local)
MINIMUM-LATENCY
MTBF
OUTPUT-INFORMATION-FLOW (local)
OVERHEAD
REDUNDANCY
RISK-FACTOR
SIGNAL-CRITICITY
SIGNAL-TYPE (local)
TYPE-OF-CONNECTION (local)

UTILIZERS Slots
member:
CONNECTED-TO-OBJECT
COST
INPUT-INFORMATION-FLOW (local)
MINIMUM-LATENCY
MTBF
OVERHEAD
REDUNDANCY
RISK-FACTOR
SIGNAL-CRITICITY
SIGNAL-TYPE
UTILIZER-TYPE (local)

PROCESSORS Slots
member:
CONNECTED-TO-OBJECT
COST
INPUT-INFORMATION-FLOW (local)
MAX-THROUGHPUT (local)
MEMORY (local)
MINIMUM-LATENCY
MTBF
OUTPUT-INFORMATION-FLOW (local)
OVERHEAD
PROCESSOR-TYPE (local)
REDUNDANCY
REQUIRED-THROUGHPUT (local)
RISK-FACTOR
SIGNAL-CRITICITY
SIGNAL-TYPE
TRANSMITTED-INFORMATION-FLOW (local)

Tab. 3-1

RAPID PROTOTYPING DESIGN

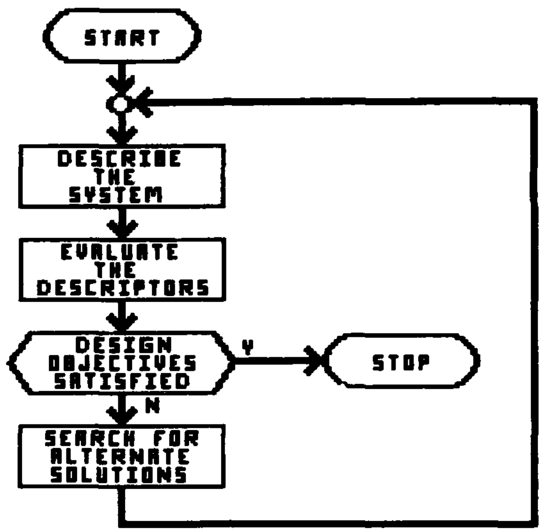


Fig. 3-1

SYSTEM DESCRIPTORS

* Total data flow	* Redundancy
* Total Throughput	* Mean time between failure
* Total memory capacity	* Integration level
* Growth Capability	* Total cost
* Latency	* Development Risk factor
* Overall efficency	* Max Risk factor

Tab. 3-2

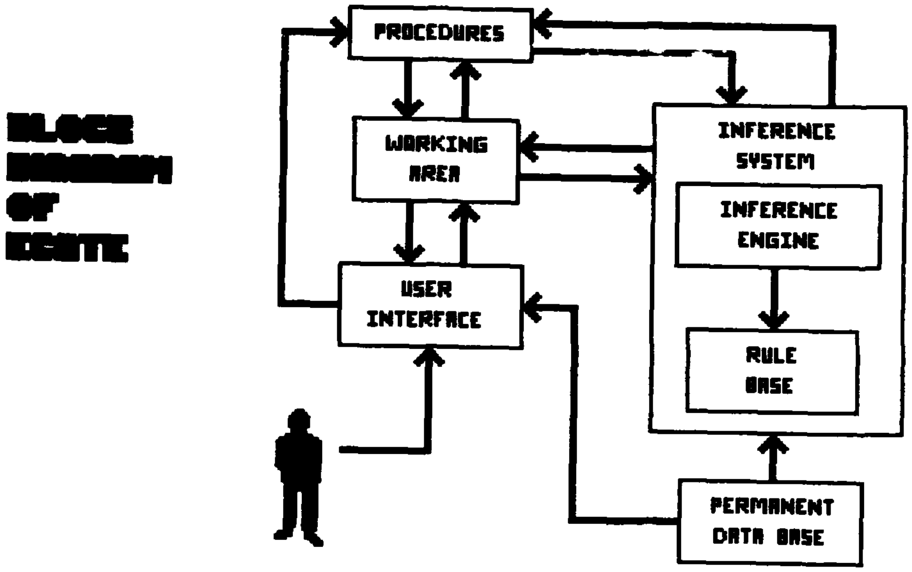


Fig. 4-1

```
(DEFUN TOT-COST (SYS)
  "Total cost computation function."
  (PUT.VALUE SYS 'TOTAL-COST
             (SUMMA 'COST (FIND-CHILDREN 'COMPONENTS SYS)))
  (BREAK-LIST (FIND-CHILDREN 'COMPONENTS SYS) 'COST 'TOTAL-COST SYS))


(DEFUN TOT-INLEVEL (SYS)
  "System integration level determination."
  (LET ((LIST-PROC (LIST-CONTROL (FIND-CHILDREN 'PROCESSORS SYS) NIL)))
  (PUT.VALUE SYS 'INTEGRATION-LEVEL
             (//
               (+ (SUMMA 'OUTPUT-INFORMATION-FLOW
                         (LET ((BIGLIST (FIND-CHILDREN 'GENERATORS SYS))
                               (PASSED-LIST NIL))
                           (LIST-GEN-CONTROL BIGLIST PASSED-LIST)))
                  (SUMMA 'INPUT-INFORMATION-FLOW
                         (LET ((BIGLIST (FIND-CHILDREN 'UTILIZERS SYS))
                               (PASSED-LIST NIL))
                           (LIST-UTI-CONTROL BIGLIST PASSED-LIST))))
              (FLOAT (+ (SUMMA 'OUTPUT-INFORMATION-FLOW LIST-PROC)
                        (SUMMA 'INPUT-INFORMATION-FLOW LIST-PROC)
                        (GET.VALUE SYS 'TOTAL-DATA-FLOW)))))))
```

Tab. 4-1

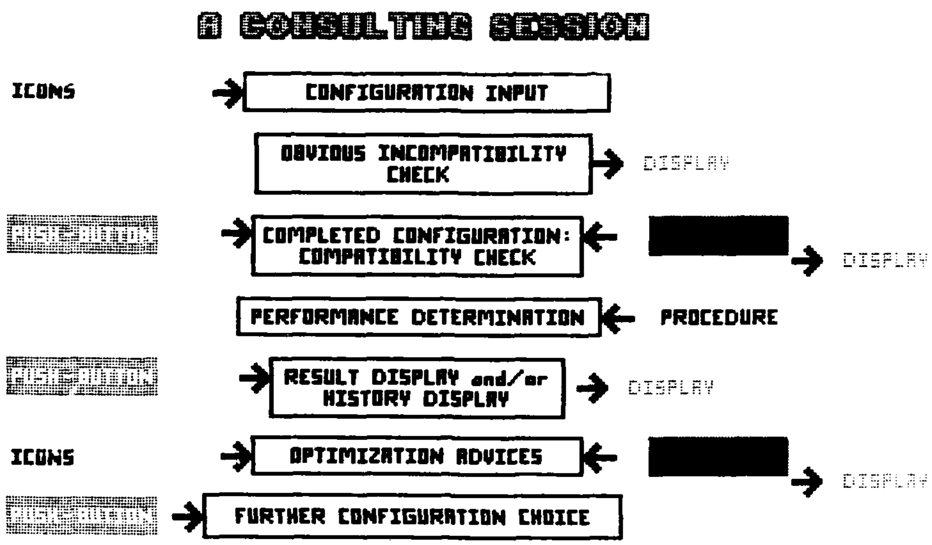

A CONSULTING SESSION
ICONS
CONFIGURATION INPUT
OBVIOUS INCOMPATIBILITY CHECK
DISPLAY
PUSH-BUTTON
COMPLETED CONFIGURATION: COMPATIBILITY CHECK
DISPLAY
PERFORMANCE DETERMINATION
PROCEDURE
PUSH-BUTTON
RESULT DISPLAY and/or HISTORY DISPLAY
DISPLAY
ICONS
OPTIMIZATION ADVICES
DISPLAY
PUSH-BUTTON
FURTHER CONFIGURATION CHOICE

Fig. 4-2

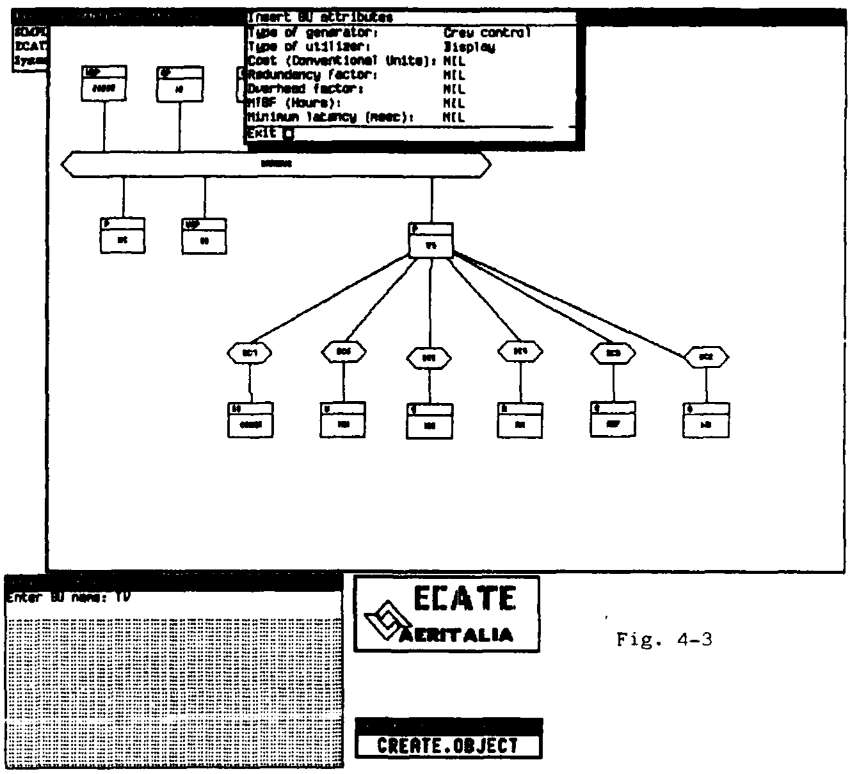

Insert BU attributes
Type of generator: Crew control
Type of utilizer: Display
Cost (Conventional Units): NIL
Redundancy factor: NIL
Overhead factor: NIL
MTBF (Hours): NIL
Minimum latency (msec): NIL
Exit
Enter BU name: TV
ECATE
AERITALIA
CREATE.OBJECT

Fig. 4-3

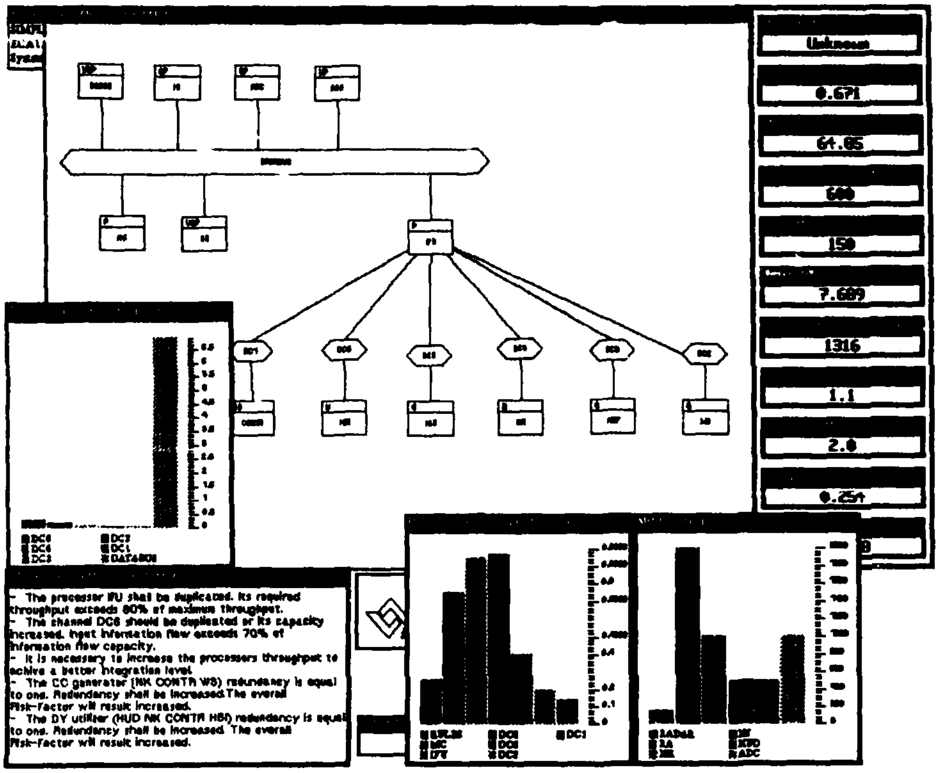

Unknown
0.671
64.85
600
150
7.689
1316
1.1
2.0
0.25
- The processor IFU shall be duplicated. Its required throughput exceeds 80% of maximum throughput.
- The channel DC6 should be duplicated or its capacity increased. Input information flow exceeds 70% of information flow capacity.
- It is necessary to increase the processors throughput to achive a better integration level.
- The CC generator (NK CONTR WS) redundancy is equal to one. Redundancy shall be increased. The overall Risk-factor will result increased.
- The DY utilizer (HUD NK CONTR HBI) redundancy is equal to one. Redundancy shall be increased. The overall Risk-factor will result increased.

Fig. 4-4

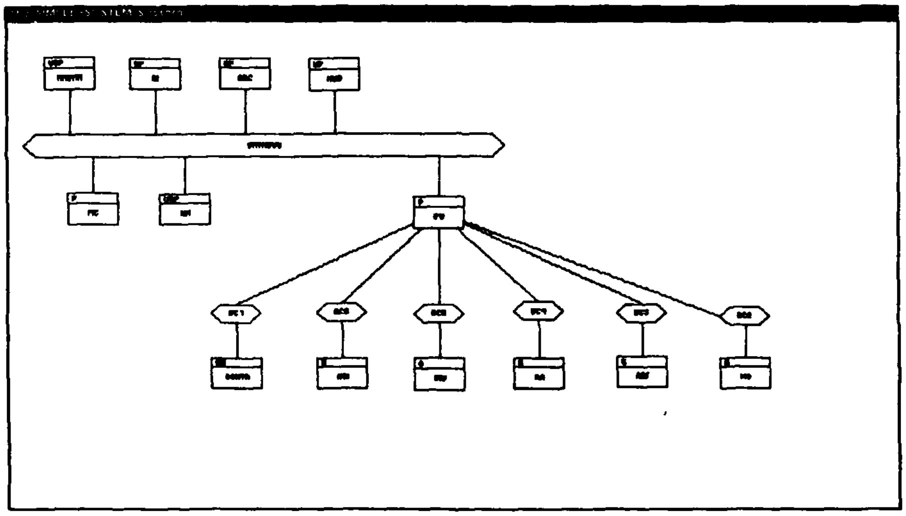


Fig. 5-1

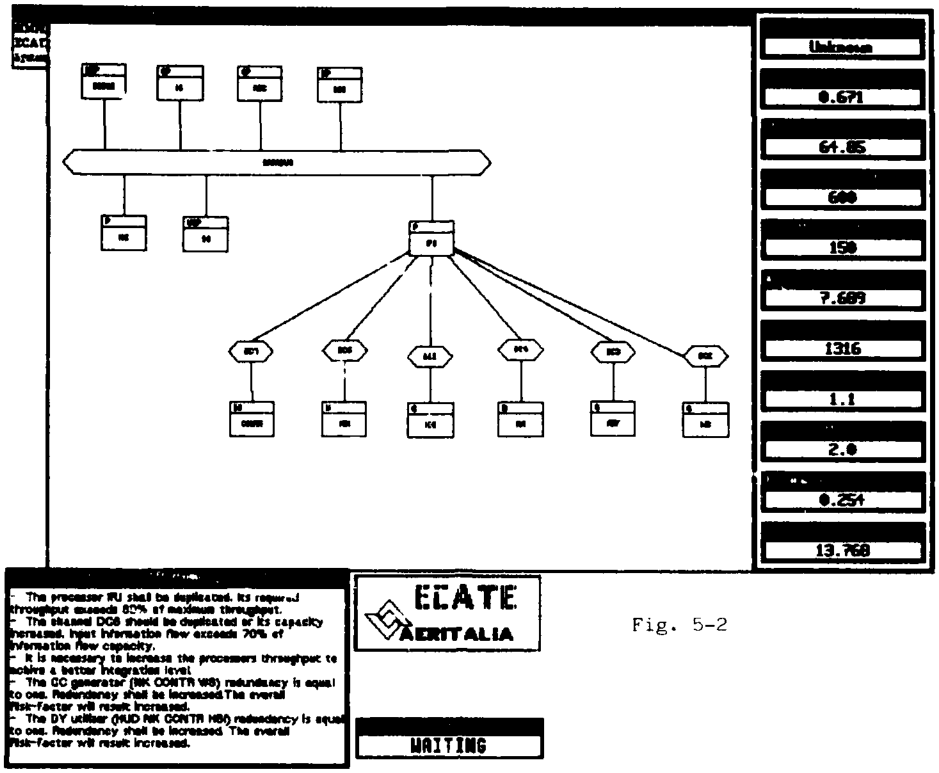

Unknown
0.671
64.85
600
150
7.689
1316
1.1
2.0
0.254
13.760
- The processor IFU shall be duplicated. Its required throughput exceeds 80% of maximum throughput.
- The channel DCS should be duplicated or its capacity increased. Input information flow exceeds 70% of information flow capacity.
- It is necessary to increase the processors throughput to achive a better integration level.
- The CC generator (NK CONTR WS) redundancy is equal to one. Redundancy shall be increased.The overall Risk-Factor will result increased.
- The DY utilizer (HUD NK CONTR HBI) redundancy is equal to one. Redundancy shall be increased. The overall Risk-Factor will result increased.
ECATE
AERITALIA
WAITING

Fig. 5-2

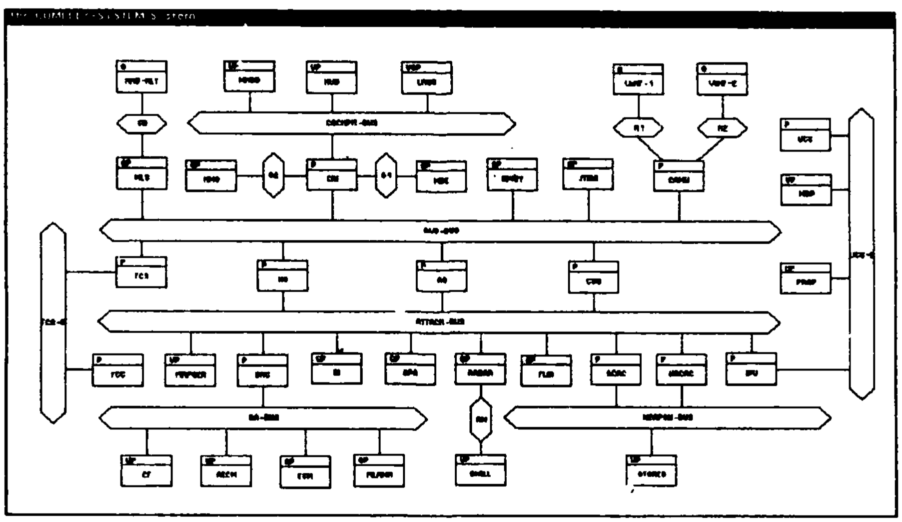


Fig. 5-3

DISCUSSION

E.Cambise, IT

Is the purpose of the ECATE system to determine the performance of an avionic system composed of building blocks with known performances, or is it to determine the performances of component building blocks to achieve the overall desired system performance?

Author's Reply

ECATE mainly addresses information processing requirements. Its intended application is the former, but it is also possible to introduce prospective performance data and assess the consequent system performance.

P.R.Walwyn, UK

Can the ECATE design tool be used to implement the "decision function emphasis" approach to system architecture design (i.e., the human interface requirement) outlined in Paper No.16, A Change in System Design Emphasis: From Machine to Man?

Author's Reply

Yes, provided a suitable model of human interface is available, ECATE can be easily incorporated in the system description. The key point is to be able to sort out the rules to operate the human interface.

THE SPECIFICATION AND DESIGN OF A FUTURE MARITIME RECONNAISSANCE AIRCRAFT

J SHEPARD

British Aerospace PLC
Civil Aircraft Division
Chester Road
Woodford
Stockport
Cheshire
UK

1. INTRODUCTON

This paper addresses the problems of providing specifications for system components of the highly integrated avionic systems of the future. The problems are discussed in the context of the avionic systems of a Future Maritime Reconnaissance Aircraft (FMRA). The need and extent of the integration of the avionic systems are discussed, as are the consequences for system definition and specification. Potential techniques for addressing these problems are reviewed. These are brought together to describe an approach which could provide the tools required. The implication of this approach for vendors and integrators is then addressed.

2. THE GROWING NEED FOR AVIONIC SYSTEM INTEGRATION

It is not too much of a simplification to say that in Maritime Reconnaissance aircraft there is not the same intimate relationship between airframe performance and avionics capability as there is in a fast jet. Provided the airframe has the capability to carry the payload required and to stay on station for the required duration its capability as an overall weapons system is largely dependent upon the capability of its avionics and weapon systems. This is shown by the fact that the Nimrod MR airframe has not been changed significantly over the years but its avionics has been updated on a number of occasions. The functionality of such a system, therefore, although not its implementation, can be considered in isolation from the airframe in which it is installed. Furthermore it has a large number of tactical sensors and communications systems to meet the needs of different roles and the performance required of it imposes high levels of integration with consequential problems in the specification, design and testing cycle.

For the purposes of specification and design, the avionics and weapon systems are broken down into major systems (Radar, Acoustics etc) each of which is then broken down into major subsystems and so on into their physical embodiment at the LRU and module levels.

The integration task commences at LRU level and progressively builds up to a functional test of the overall weapon system. The size of this task is illustrated by the fact that a system such as Nimrod has of the order of 400 avionics LRUs and 800 avionics LRU interfaces, some simple but some involving a large degree of interactions between the subsystems. The architecture of a typical system of this type is shown in figure 1 and a typical specification, design and integration testing cycle is shown in figure 2. This could take of the order of 5 - 10 years.

3. FUTURE DEVELOPMENTS

Consideration of the future threats indicates that greater performance will be required of future avionics systems. Greater sensitivity in the sensor heads will produce more data but it is unlikely to result in data better than that currently available and it may well be worse. To maintain or increase mission success rates with these quantities and qualities of data will require more sophisticated processing, both within systems and across systems The ensuing need for a higher level of integration will, at its most extreme, result in the kind of system architecture shown in figure 3 where processing functions can be shared. The increase in the number of sensor systems but reductions in the size of elements of such systems e.g. by the use of VHSIC, could lead to elements currently specified as LRUs being implemented as single card modules and sharing an enclosure with modules from other systems. Even if not fully realised the flexibility offered by some of these developments will have similar effects. For instance the moves towards general purpose computing elements e.g. the Transputer, offer cheap, reconfigurable processing which can be extended in a way which is functionally transparent to the software using it, except in performance terms. The system requirement, and the software element of it, may then be designed without knowledge of the hardware embodiment and hence functional interaction between components must be explicitly specified.

These developments will pose in stark terms the type of problems faced today by the specification and integration of avionic systems. These are:-

- the need to provide an unambiguous definition of the functionality of the system and subsystem,

- a definition of the dynamic interaction of components within a system,

- a definition of the dynamic interactions of systems across their interfaces.

4. POTENTIAL TECHNIQUES TO ADDRESS THESE PROBLEMS

Almost all of the potential techniques considered have their origins in software, where the abstract nature of the product leads to attempts to define the requirement and deliverables as explicitly as possible.

4.1 Functionality

Functionality is the main area which has been addressed. The techniques developed have some combination of the following ideas :

- diagrammatic representation,

- unambiguous notation,

- functional decomposition,

- data modelling (for data bases),

- process modelling.

The techniques may also incorporate a set of procedures by which the techniques are applied. The degree of interpretation required to implement the functionality defined is dependent upon the precision or degree of elaboration used by the techniques and procedure. Since many of these techniques are intended for use in transaction processing they lack a means of temporal representation. They also range from fairly abstract means of representation to techniques such as MASCOT which maps ideas onto language and hence leads directly into tools such as CONTEXT which converts MASCOT into CORAL. The type of technique required for the FMRA must provide an unambiguous representation which can indicate temporal relationships. It must support the specification, design and testing cycle.

4.2 Dynamic Representation of System Components

The idea of executable specifications and prototyping also has its origins in software. The production of a functional requirement document leaves open the question of whether the requirement, as it lies on the page of the specification, or the initial paper design, in response to that requirement, are viable individually and whether the design is an adequate response to the requirement. It is an attractive idea to write the specification in software, which can be executed, and hence proved viable, and to write a prototype of the software design which can be proved viable in its own right but can also be compared with the executable specification for verification.

This is an extremely difficult task for a time dependant, highly integrated system. The types of language commonly available do not lend themselves to the functional modelling of the requirement and hence a fair degree of effort is required to actually construct a model. Consequently the idea of the frequent change needed to refine the requirement is daunting. The same is also true of prototyping, except the prototype is invariably more detailed than the requirement hence the work involved in the changes is the greater. The technique required for the FMRA must be capable of adquately representing the system components of the requirement specification, the interactions between these, and must also allow the definition of the components and the interactions to be changed reasonably easily. The definition thus established can be developed by more conventional means.

4.3 Dynamic Interactions Across Interfaces

Interface design is generally agreed to be both difficult and important but almost all the effort has been directed at providing improved or more detailed documentation. By its nature this means of providing the interface specification is static and defines the detail of an interface rather than indicates the way in which the detail is to be used. It is thus open to more than one intepretation because it is a detailed static definition of a boundary representing complex, dynamic and time-related interactions between functions either side of the boundary.

The technique developed to represent the dynamic representation of system components would produce the starting point for such an interface definition but it would not be detailed enough. The technique required for the FMRA must not only represent the system components on either side of the boundary adequately but must also be able to represent the data exchange characteristics.

5. TECHNIQUES FOR THE FMRA

5.1 Functionality

A system design method with which BAe has some experience is Controlled Requirements Expression (CORE). This is a technique which produces a temporal representation of processes with data passing between them. Each process has a process description and each datum a definition. It uses the technique of functional decomposition and is capable of being continued until software is produced. It is necessary for the method selected to deal with the fact that a weapon system developed by a Prime Contractor would consist of functions provided by Government, functions developed by the Prime Contractor and functions defined by the Prime for a sub-contractor.

All of these functions can be defined, to a greater or lesser degree of detail as appropriate, using CORE. The GFE functions must be defined to a level of detail which allows their effect on the overall weapon system and its major system be determined. The subcontracted functions must be defined to a level of detail which allows a proper degree of control over the subcontract. The self to self function must, of course, be defined to a production level of detail. CORE allows the definition to be stopped on different levels of detail and still produce an adequate definition. The problem, both for current and future systems, is establishing the boundaries of the system.

Figure 4A shows how a current LRU based system would be broken down in this way. Figure 4B shows an equivalent diagram of a future system indicating that the method can be used to produce a requirement definition for a future system. This definition is however a static one.

5.2 Dynamic Representation of System Components

The definition of a major system, such as Radar, in CORE would produce functional descriptions of the major blocks (see figure 5). Figure 6 shows an expansion of the tracking block where the "track control" process description is given. This type of description in which the process is described in the form rules relating input data, output data and intermediate results, is evocative of a Knowledge Based Systems using production rules. The essence of such systems is the ease of changing the knowledge compared to a system written in a conventional language where the knowledge is implicit in the structure of the software. Using an architecture known as a Multiple Blackboard architecture a model of the system could be built up from the processes defined by CORE. Each major system, e.g. RADAR, could have its own blackboard upon which the system functions, e.g. signal processing or tracking, read data intended for Knowledge Sources (KS) e.g. track control, and write data from KS e.g. transmit track.

These blackboards are themselves the KSs for a system blackboard on which the major systems read and write data (Figure 7). This type of approach meets the characteristics stated. These are:

a) it is capable of adequately representing the components in CORE

b) it is capable of adequately representing the interaction in CORE then in KBS

c) the definition of component and interaction can be changed reasonably easily - this is an attribute of a KBS

Since the system is defined in terms of functions, boundaries can be defined which produce (arbitrary) groupings and the interfaces between them. Such a grouping could be GFE or subcontracted functions and changes to such functions could be reflected throughout the model. The model is thus capable of testing the effect of changes in the subsystem functions and the overall system regardless of the physical boundaries of the subsystem (Figure 8).

If a great deal of detail is required in the model it will be necessary to develop a model of the major systems, the multiple blackboard architecture lends itself to this line of development.

5.3 Dynamic Interactions Across Interface

From the foregoing discussion it is apparent that the same general approach would also be appropriate for interactions across interfaces. An interface definition is an evolving definition. It starts with sparse detail and gradually is developed until it can become so detailed that it is not fanciful to describe it as encrusted (and virtually impossible to verify or validate). It also combines functional data on what systems do with data on protocol and procedure; the CORE definition will provide the former, the traditional ways of defining protocol and procedures will produce the latter. In considering protocol and procedures it is only necessary to recall that these are other names for rules to see that the CORE/KBS approach is appropriate both for existing LRU-based systems and for future systems (Figures 9 and 10).

It should be noted that interface definition is required between all functional groups whether GFE, subcontracted or in-house developed.

6. A Coherent Approach

These three types of task can come together into a coherent approach (Figure 11). The overall system can be defined in CORE to the point at which groupings are established. These can be modelled using the CORE/KBS approach as can the interfaces between these groupings. After this point GFE functions are only modelled in the detail required to support the other functions. This means in full detail for the interface models. The CORE definition of the other functions and subsequently the CORE/KBS models, are developed to the level of detail required to allow the definition of subcontracted functions and their interfaces to be delivered to the subcontractor. After this point, subcontracted functions are modelled to the level of detail required to support the other functions and allow control of the subcontracts to be maintained. This involves full detail of the interface models.

7. IMPLICATIONS

The preceding discussion deals only with issues arising from the need to provide an effective means of specifying and designing a highly integrated system. There is evidence that subcontractors are reluctant to accept as specifications anything other than conventional documentation. The existence of in-house methods; the investment in trained personnel; the flexibility afforded by the in-house decision as to how a requirement specification will be used is the basis of a design; these are all factors which will hinder the adoption by a Prime Contractor of a specification and design approach which is based on handing over software models as part of a requirement specification to be met. More particularly this approach brings in its wake the question of standardisation on computer, operating system and software tools. The approach is, in any case, fairly ambitious in its demands upon the KBS architecture (still in development) and the software tools needed to produce such models (currently KBS shells are special to type). Whether the approach achieves a realisation is largely dependent upon whether the needs of future platforms demand such highly integrated systems.

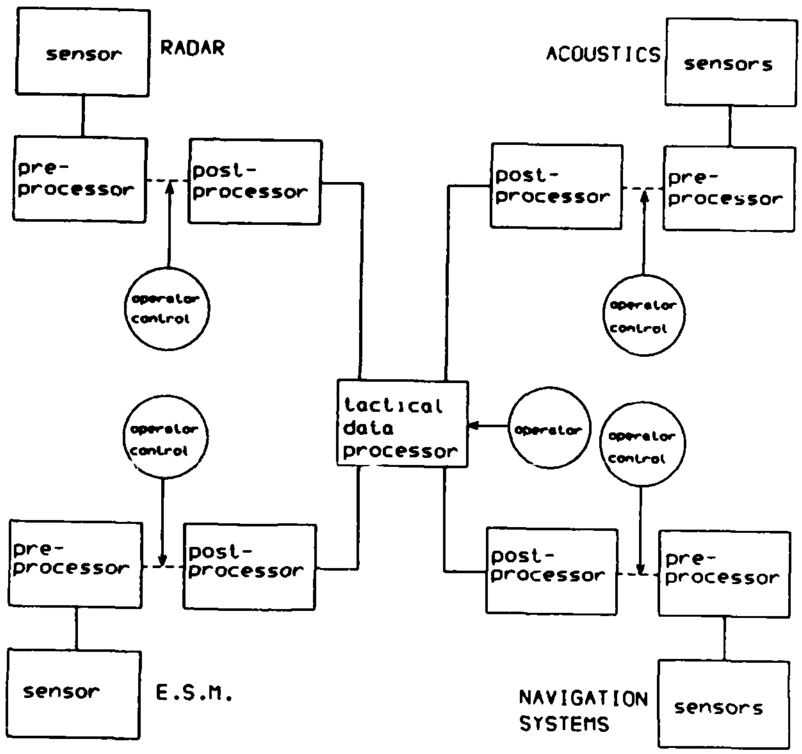


FIGURE 1

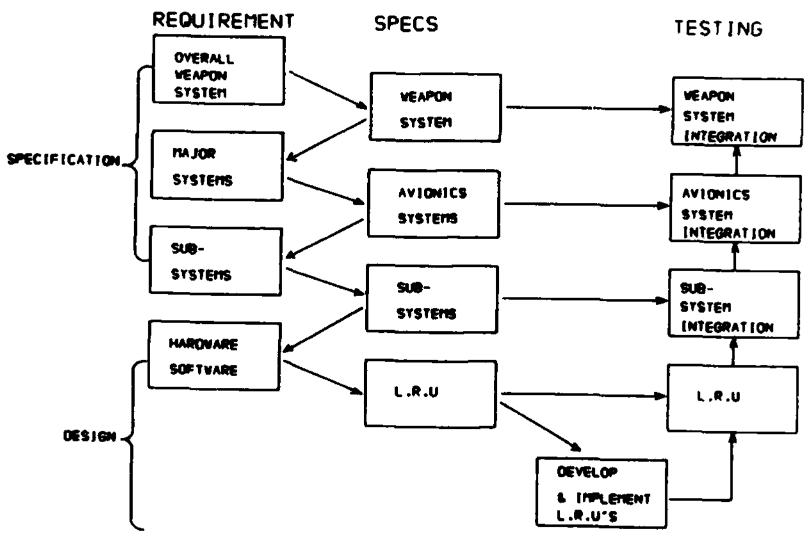


FIGURE 2

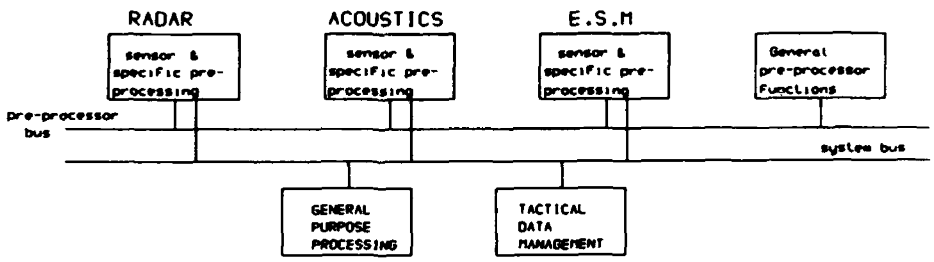


FIGURE 3

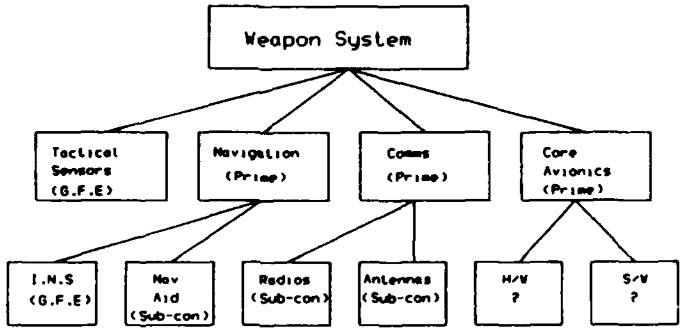


FIGURE 4A

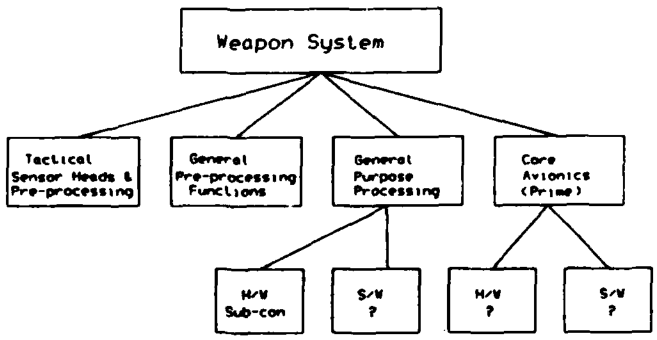


FIGURE 4B

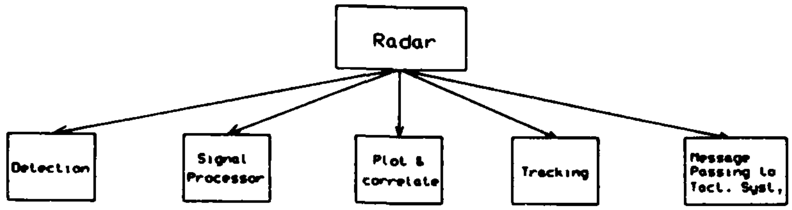
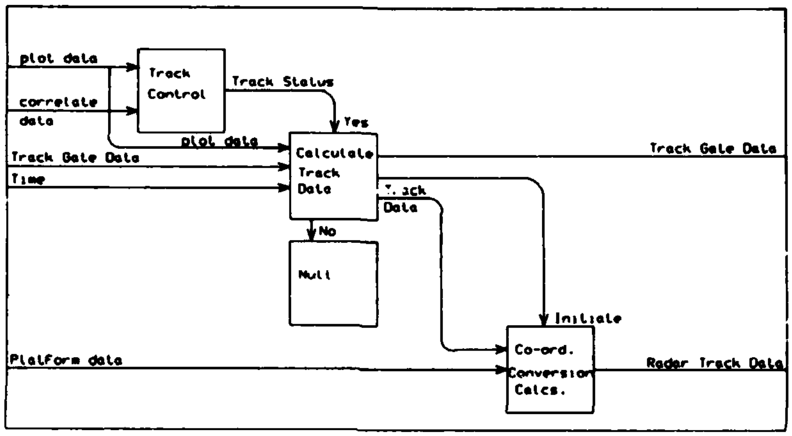


FIGURE 5



Process definition
Track Control:- track status is "Yes" if both plot and correlate data are available and if plot data occurs 3 times consecutively.

FIGURE 6

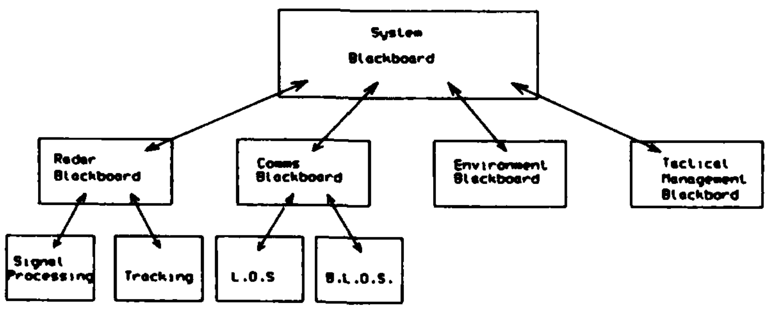


FIGURE 7

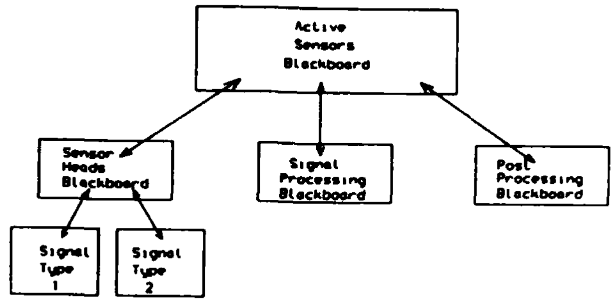


FIGURE 8

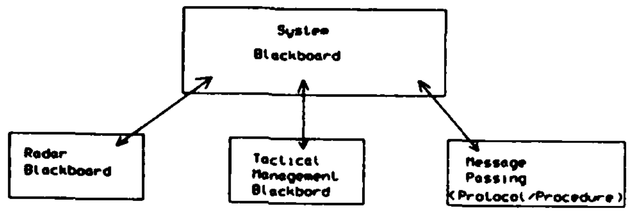


FIGURE 9

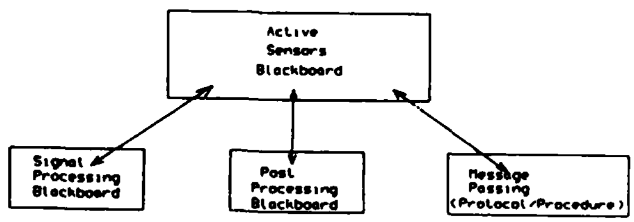


FIGURE 10

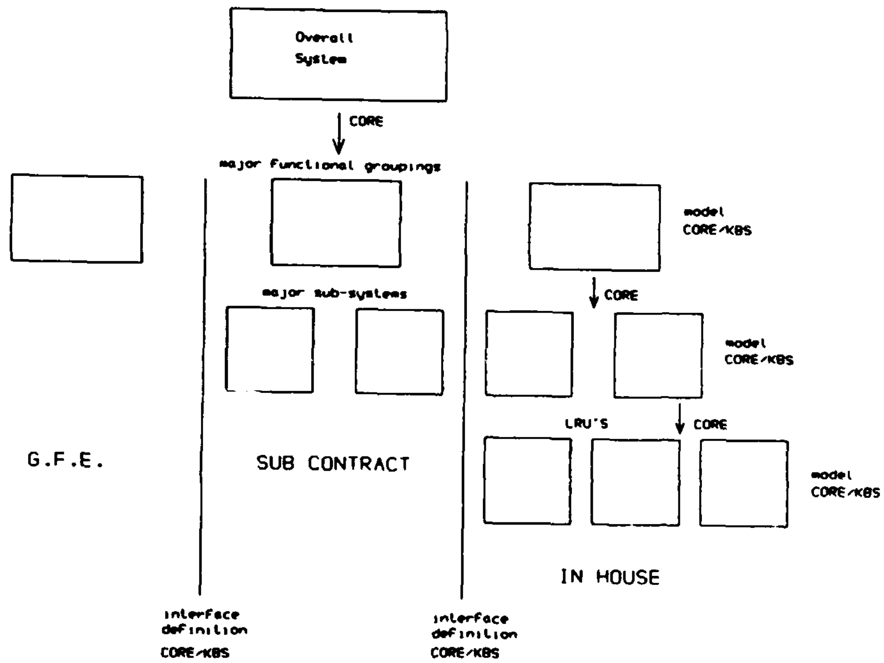

Overall
System
CORE
major functional groupings
model
CORE/KBS
major sub-systems
CORE
model
CORE/KBS
LRU'S
CORE
G.F.E.
SUB CONTRACT
model
CORE/KBS
IN HOUSE
interface
definition
CORE/KBS
interface
definition
CORE/KBS

FIGURE 11

A STRUCTURED APPROACH TO WEAPON SYSTEM DESIGN

by

H. M. MALLEY, N. T. JEWELL, R. A. C. SMITH

BRITISH AEROSPACE, MILITARY AIRCRAFT DIVISION,
WARTON AERODROME, PRESTON,
LANCASHIRE, ENGLAND.

SUMMARY

The design requirements for future airborne weapon systems show an increasing drive towards improving their overall performance and flexibility whilst at the same time reducing the total weight and the resulting cost. These requirements generate the need for much closer integration of the subsystems which make up the total weapon system. This approach to total systems integration, forces the weapon system designer to look for improved design techniques which are capable of coping with the complexity and high interdependence of the various functions involved.

This paper describes a structured approach to the design of highly integrated weapon systems of the future. The approach was used in the successful design and development of the avionics system for the UK Experimental Aircraft Programme (EAP) demonstrator aircraft. Brief descriptions are given of the EAP systems, the main system design tools used, the activities carried out during the systems design process and the management and control procedures adopted. The paper concludes with a series of observations highlighting some of the findings of the project and providing pointers to the design of future weapon systems.

1. INTRODUCTION

The prime purpose of this paper is to describe a structured approach to the design of a weapon system which British Aerospace (BAe) were able to develop and prove during the design of the avionics system for the Experimental Aircraft Programme (EAP) demonstrator aircraft. This aircraft first flew in the United Kingdom in August, 1986. Brief descriptions are given of the EAP systems, the main system design tools used, the activities carried out during the systems design process and the management and control procedures adopted. In addition a series of observations highlighting some of the findings of the project and providing pointers to the design of future weapon systems are given.

What is a structured approach? - A structured approach can be defined as a methodical or step by step process involving the progressive development of concepts from the start of the design process to its completion. This contrasts greatly with the ad hoc methods which have been used in the past and have been found deficient in such areas as the lack of definition or the late discovery of design or specification problems leading to costly modifications. In no way does the structured approach remove the need for engineering judgement or skill, it does however highlight the need to make engineering decisions at the appropriate time to allow the design process to proceed.

Why a structured approach? - As technology continues to advance so the requirements demanded of the next generation of weapon systems also increase. This increase is two-fold; improvement in the performance of existing requirements such as detection ranges or navigation accuracy and improvement in operational flexibility by the combination of requirements such as air combat and electronic counter measures into one vehicle. In this age of the ever increasing demand for the use of new technology, it is essential that its use is kept within the bounds of what can be afforded. In the case of an airborne weapon system, weight is considered to be a very important cost driver. The heavier the avionic equipment, the greater the demand it makes on the aircraft, i.e. airframe, engines, fuel, etc., which leads to a heavier, costlier aircraft. There is therefore a very strong requirement to minimise the weight of any airborne weapon system.

In the past it was considered acceptable for individual system elements to stand alone. This often led to some duplication in the provision of sensors, processors or displays. The demands for minimising the total system weight removes any possible duplication of the system elements and leads to a highly integrated weapon system where some equipments may undertake several functions. It is this complexity and interdependence of the various functions involved which forces the weapon system designer to look for improved design techniques.

It was therefore considered that the following features should be embodied in the processes to be used for the design of the EAP avionics system:-

- a step by step approach which progressively develops the design rationale and provides a capability for planning the execution of the design and for monitoring its progress.

- a precise, consistent and unambigious way of expressing system requirements at all levels.

- a means of applying checks at different stages of the design life cycle to detect errors of specification or design in order to assure the design quality.

- an ability to demonstrate that the requirements have been met in order to provide traceability of the requirements.

The above features constitute the structured approach which was adopted by BAe and which are discussed in more detail in this paper.

2. EAP

While EAP has been created relatively quickly, its origins go back at least ten years. During this period, engineers at BAe worked on various studies for a new fighter aircraft incorporating twin engines, delta wings, canards and both single and twin fins. Some of these studies were undertaken in collaboration with other Companies. In 1979 a proposal for a European Combat Fighter (ECF) was put to the British and German governments jointly by BAe and Messerschmitt-Bolkow-Blohm (MBB), while in 1980 a slightly modified design for a European Combat Aircraft (ECA) was prepared by BAe, Dassault-Breguet and MBB and put to their respective governments. Unfortunately the governments were unable to reach agreement on a common set of requirements.

During 1981 BAe continued its studies and defined the P110 project which involved the UK Avionics Industry in agreeing a weapon system architecture and producing equipment specifications. At the same time MBB were working up their TKF90 project which was very similar to the P110 project. Therefore in April 1982, BAe and MBB together with Aeritalia who had previously co-operated to design and build the Tornado aircraft, agreed to investigate the possibilities of producing a joint specification to meet their individual national requirements. The resulting Agile Combat Aircraft (ACA) was unveiled in mock up form at the 1982 Farnborough Air Show and the UK government announced that they would provide support for a demonstrator aircraft EAP which would be flown at the 1986 Farnborough Air Show.

It was anticipated that two demonstrator aircraft based on the ACA design would be built, one in Great Britain and one in Germany. During 1983 a limited systems fit was agreed for the aircraft and due to the tight timescale of the project, equipment specifications were produced, put out to tender and Suppliers selected, very much in advance of the carrying out of a detailed system design. Unfortunately as a result of the German and Italian Governments decisions to withdraw at the end of 1983, work on the German aircraft did not proceed. However the chosen equipment suppliers from the three countries accepted BAe's invitation to continue with the design, build and supply of the numerous equipments required, without charge, for the single demonstrator aircraft.

In obtaining the agreement of the UK government to provide support for the demonstrator aircraft, it was necessary to agree the objectives which would be demonstrated. The areas chosen covered the fields of aerodynamics, structures and materials, and systems and involved the development and demonstration of procedures necessary for the design, manufacture and test in these areas which were considered relevant to a future fighter aircraft.

This paper concentrates on the work carried out in the design and development of the avionics system involving the use of the MIL STD 1553B data bus and the modern electronic cockpit which were the responsibility of the authors.

3. EAP SYSTEMS

The EAP has three major electronic systems: Flight Control System, Utilities Services Management System and Avionics System, the latter comprising communications, navigation and displays and controls subsystems. A simplified system architecture is shown in Fig. 1.

3.1 Flight Control System

The EAP has a full authority, digital fly by wire system to provide artificial stability and the necessarily complex control functions. This system is based on the Jaguar Active Control Technology aircraft which was the first aircraft to use fly by wire for flight control without mechanical back up. The system controls up to 13 surfaces simultaneously. The four identical flight control computers host the flight software which enables the pilot to fly the unstable aircraft and provides carefree manoeuvrability and increased agility. The computers also house software for failure management, reversion logic and built in test. The computers receive inputs from the four aircraft motion sensors, four attitude detectors, two air data computers and pilot inceptors and provide the outputs to the control surfaces. In addition they provide air data information to the utilities and avionics systems via the two MIL STD 1553B data buses and air date, attitude and heading to the reversionary instruments.

3.2 Utility Services Management System

While not originally claimed as a technological demonstration feature, the EAP adopted an integrated computing system for the control and management of the aircraft utility systems. The system comprises four system management processors connected to a dual redundant MIL STD 1553B data bus. The bus control function is embedded in two of the processors. The main utility systems which are controlled by the processors are as follows:-

- fuel management and fuel gauging
- hydraulic system control and indication
- undercarriage indication and monitoring, wheel brakes
- environmental control system including cabin temperature control
- engine control and indication
- secondary power system
- LOX contents, electrical generation and battery monitoring, probe heating

The main benefits of this type of system for a high performance aircraft are a significant reduction in installed weight and operating costs, and a large improvement in availability. It also provides a simple interface to the avionics system in particular for the cockpit electronic displays and controls, this being one of the original drivers in evolving the system.

3.3 Avionics System

It was accepted that for the EAP, the avionics system would be a sub-set of the weapon system proposed for the ACA. This was called the 'Core System' and provides the essential features to fly a high performance aircraft namely navigation, communications and display and control functions. Transmission of data between the subsystems is via a dual redundant MIL STD 1553B data bus. This greatly reduces the amount of wiring required in the aircraft and simplifies the development and on-aircraft testing.

The navigation subsystem comprises an inertial platform with its own self-contained navigation processor and a TACAN and radar altimeter which share the same remote terminal.

The communications subsystem comprises a standard V/UHF radio and an emergency UHF radio. The control of this equipment is through an integrated control and management unit which also provides a voice warning facility. The latter supplements the normal aircraft warning system.

The displays and controls subsystem demonstrates several new technologies. Two identical waveform generators form the heart of the subsystem and are each capable of driving the three multi function colour displays and the wide angle holographic head up display. They also provide the bus control and executive control functions for the avionics system. Mission data such as waypoints, TACAN beacons, communication channels etc., are inserted by the pilot via the manual data entry facility mounted on the left hand glareshield. This information together with raw control data from the controls mounted on the consoles, throttles, control stick and displays is processed in two identical cockpit interface units prior to being transmitted on the avionics data bus. A cockpit lighting controller undertakes the task of monitoring the light sensors distributed around the cockpit and continuously regulates the power supplied to all the displays and controls to provide optimum illumination and display contrast at all times.

4. DESIGN TOOLS

In parallel with the work being carried out on fighter aircraft studies during the 70's, BAe put considerable effort into examining ways of improving the techniques used for designing systems and also into the newer system technologies. Two specific areas which showed promise and were pursued with the support of the UK government, were:

- means of improving the production of airborne software in terms of productivity and quality
- investigations into the implementation of a MIL STD 1553B databus together with an all 'electronic' cockpit including the multi-moding of displays and controls.

The former led to the development of an approach called Semi-Automated Functional Requirement Analysis (SAFRA) while the latter led to the production of the Active Cockpit. Both of these tools were used to support the design of the avionics system for EAP and are described in the following paragraphs.

4.1 Semi-Automated Functional Requirements Analysis (SAFRA)

In examining ways of improving the productivity and quality of airborne software it was shown that the biggest improvements would be obtained by the ability to find and eliminate errors at as early a stage as possible in the software life cycle. This led to the development of the approach called SAFRA which in particular addressed the lack of method, lack of visibility, lack of consistency and resolution of ambiguities in producing software requirements. Just as this method was applied to software requirements it was shown that it could be applied to the establishing of system requirements and was therefore also adopted for this latter purpose.

The SAFRA approach encompasses a number of methods and tools which support the various stages of the system/software life-cycle. At the heart of SAFRA is a method called COntrolled Requirements Expression (CORE) which is used to produce system and software requirements that are unambiguous, consistent and complete. The method is based on the progressive decomposition of high level requirements in a logical and consistent manner until a level is reached where the requirements are expressed in sufficiently precise detail to allow hardware and software design to commence.

Each level of decomposition consists of a number of logical steps, eleven in all, which when applied to a higher level requirement produces the lower level components of the requirement. These steps can be collectively summarised as information gathering, establishment of relationships and the verification of relationships. The information derived at each level of decomposition is presented in diagrammatic form known as CORE diagrams. These diagrams use a precise unambiguous notation which can be checked for consistency and completeness across the whole of the systems requirement.

To assist in the production of CORE diagrams, a work station was developed which enables diagrams to be entered at a high-resolution graphics terminal and edited as required. It also provides a multi-user database in which diagrams are stored and a hard copy facility using a printer-plotter. Some automatic on-line checking of the diagrams for consistency is undertaken as they are being entered.

By producing the requirements in an unambiguous form in a computer database it is possible to check the data for consistency and completeness. This was done using PSL/PSA (Problem Statement Language/Problem Statement Analyser), which is a product of the ISDOS project of the University of Michigan, although other similar products such as EPOS are now available. The CORE notation is automatically described in PSL in a consistent manner and stored in a new database. PSA is then used to provide checking and analysis of the database in numerous ways. When all the checks have been satisfactorily completed at each level of decomposition, the CORE database is made read-only to allow the next stage of the design to proceed.

4.2 Active Cockpit

As the result of its continuing development studies, BAe gained considerable experience in the operation of active cockpit facilities and demonstrated their great importance in providing information on the man-machine interface to the system design process. In effect this facility provides a means by which rapid prototyping of ideas can be tested and developed with full operator interaction.

The Active Cockpit consists of a wooden shell representing the actual cockpit. The displays and controls are positioned to the best information available, use being made of commercial items wherever possible. Initially static displays are assessed for the development of the moding and formats. These displays are then driven dynamically in a representative manner together with other simulated functions such as engines, hydraulics, fuel etc., to fully exercise the cockpit displays and controls. To allow assessment under realistic flight conditions an outside world simulation system is provided. In addition a comprehensive fault injection system is used to allow assessment of the pilot/cockpit interface under single or multiple system failure conditions.

5. SYSTEM DESIGN ACTIVITIES

The life-cycle stages in the design and development of a typical system are illustrated in figure 2. These can be grouped under three main headings namely system design, system implementation and system test. As defined system design covers the stages of activity from the establishment of the initial high level system requirement through several levels of decomposition which produce the detailed requirements including hardware/software partitioning to the production of hardware and software specifications. System implementation covers the stages from the availability of specifications through to their realisation in either hardware or software. System test covers the stages of testing starting with individual equipments and software modules and building up individual elements to subsystems and finally integrating these to form the total system.

The following paragraphs describe in more detail the various activities undertaken during the system design process. These activities are considered under four major headings: design planning, data gathering, functional analysis and partitioning.

5.1 Design Planning

Design planning focused on the production of a design route map which is illustrated in Figure 3. This route map showed how each of the design teams intended to produce formal design documentation. Each system route map was also required to show the dependence of one design task on another so that a sensible programme could be derived. Technical details relating to the use of CORE were also included e.g. the number of levels of design decomposition were considered and documented, and naming conventions for data items were included. To ensure proper interfacing and integration of the main aircraft systems, the points at which formal design reviews would be held, were also defined. This ensured that the system design was properly integrated thereby avoiding the need for major and time consuming design revisions late in the programme.

The route map also defined a list of formal documents which were to be produced and delivered to other organisations. For example the systems design team were tasked with providing the software team with software requirements and software specifications: the scope and depth of each of these documents were also defined.

5.2 Data Gathering

A data gathering activity occurs in any project either as a conscious or an unconscious action. Probably the most commonly accepted need at the start of any project is to obtain a basic design requirement for the aircraft. This is usually provided by a formal requirement from the customer such as an air staff target or requirement. In the case of EAP no such requirement existed and it was therefore necessary for the aircraft companies to produce a similar document. During the definition of the ACA project, attempts had been made to establish an air vehicle specification. It was therefore possible to modify this specification to reflect the intended standard of the EAP. This was then used to establish the areas of high technology to be demonstrated which formed the basis of the agreement with the UK government for providing some of the funding for the project. Thus some constraints and guidelines were formally agreed at an early stage of the project.

To complement the basic 'customer' requirement, the design teams produced a set of preliminary systems descriptions. In effect these descriptions were based on design which embodied experience resulting from rig demonstrator programmes, advanced aircraft studies, and a knowledge of the state-of-the -art in equipment and systems. They were not validated design documents and subsequently were not maintained as formal design statements. These descriptions were formally reviewed by the project management early in 1984. It is of significance that subsequent to the issuing of the systems descriptions no further major changes in requirements were permitted until this initial design had been implemented. This meant that a relatively stable set of design objectives were available.

A further stage in the data collection was the production of a set of principles and philosohpies which would guide the subsequent detailed design. A typical example was to derive the concepts for displays and controls moding for ergonomic single crew operation. Also important at this stage was the prototyping of design ideas, and in particular the use of the active cockpit rig discussed in Section 4.2. This facility allowed pilots and designers to assess design concepts. Because of the lead times needed to produce such a facility the active cockpit did not provide validated design data during the data collection phase, however all of the information to be used on the rig was also made available to the system designers. In this way rig based revisions were fed into formal design documents at a later stage, but in advance of the design freeze prior to subsystem testing.

Finally an area of data collection which is either overlooked or is paradoxically taken to be a comprehensive statement of a design is that of pre-defined functions. On EAP several of the required functions could be obtained by the use of existing or slightly modified equipments, typically, TACAN, IFF and inertial navigators. The non-standard aspects of these of these equipments was not related to the functions they provided but to the way in which the equipments would be interfaced. Thus by defining in CORE the functions and data requirements for these equipments, constraints were placed on the design. It also enabled significant pieces of the system 'puzzle' to be put in place quickly and used to drive out other less well defined system requirements. It should be noted that this is a technique which is generally applicable. For example the weapons fitted to a new aircraft are rarely completely new and hence the accuracy, functions and data requirements of many weapons will be known at the initial design stage and should therefore be used to drive the design of the interfacing equipment.

5.3 Functional Analysis

Functional requirements analysis was the first formal design stage of EAP and used the CORE systems design tool. Because CORE is essentially a method for documentating, analysing and validating a design, all the preceeding informally derived data was first entered into the system CORE design database. As an example the initial step in CORE is to propose the main system functions and then to postulate their data requirements. This is formally referred to as tabular entries, and much of this data was available from the previous stage of data gathering. However, because of the diagrammatic nature of CORE and the fact that the design data is electronically stored, it could be validated for functional completeness and consistency using the PSL/PSA tool e.g. for each data source there must be a user.

Initial system moding was also proposed at this stage. Each function was described in terms of its various modes or states so that the control data needed to enter/exit a particular mode, and the specific data produced in that state were defined.

Space does not allow all of the separate and formal design steps associated with CORE to be discussed here, however the benefits arising from this formalism were numerous. Direct benefits included automatic checking of gross system inconsistencies, automatic production of interface control documents for the major systems, and a clear definition of the bus control transaction table requirements. Of less importance at the functional design stage but of considerable benefit later during systems testing was the ability of the various disciplines (test, system design, and software), to quickly trace a fault, isolate the cause, and correct the problem without inducing additional problems elsewhere in the system. The formalism and visibility provided by the CORE design method also dramatically reduced the number of errors which were detected during the final stages of system integration testing.

To provide sufficient detail, the design of the avionics system was completed in three separate stages.

- The first stage was the production of a functional statement known as the system functional requirement document which described the system functional requirements without reference to mechanisation or location and was derived from the aircraft specification, systems descriptions and the principal and philosophy documents which had been produced. It also defined the information flows to and from the avionics system.

- The second stage involved the decomposition of the system functional requirements to establish subsystem functional requirements. This provided more detailed definition of data attributes, transfer rates and interface details such as word formats. It also included the overall sequence of operation of the various functions and their iteration rate. This resulted in deriving the data interfaces between subsystems/equipments.

- The third stage involved further decomposition of the subsystem functional requirements to produce the detailed hardware and software requirements. The software requirements were in fact produced in two phases. As soon as the data interface requirements, sequence of operations and iteration rates were available from stage two, it was possible to define the software schedules and the overall structure of the code so that the software basic design could proceed. The next phase was to add the details of the process algorithms. This enabled the software task to be completed by coding the algorithms which were in principle inserted into the basic design as process modules.

5.4 Partitioning

It is clear that at some stage there must be a transition from the purely functional design requirement to an implementable system specification. For the most part this transition was done during the second stage of the functional analysis. The main functions were partitioned into equipments, and in the case of multiprocessor units, they were allocated to specific processors. However, this procedure was not a simple single step action. The main systems were partitioned at the first stage i.e. functions which were to be carried out within the flight control, utilities and avionics systems were defined. Subsequently each of the functions allocated to these systems were partitioned to appropriate subsystems. For example, avionics partitioned functions to the navigation, communication, displays and controls subsystems.

This partitioning process took account of such system technical requirements as minimising data bus load requirements and data latency and providing adequate failure mode handling. In addition project considerations such as minimum weight and the use of pre-defined equipments were also taken into account.

It was found quite acceptable in certain instances to split up functions amongst the various equipments or processors or to combine several functions in a single equipment or processor. The main requirement was that the design was clearly documented.

Two examples are given to illustrate the above mentioned forms of partitioning.

- The EAP avionics system included an intelligent voice warning system which involved two fundamentally different disciplines, the voice audio generation and management and the logic associated with recognising a failure and associating this with a specific warning requirement. The waveform generators were in effect natural hosts for a good deal of the system data and hence were already sourced with the data needed for warning generation. Conversely the communications management system naturally provided audio generation, audio mixing, etc. Therefore the voice warning generation function was partitioned to the communications management unit, and the warnings handling logic was partitioned to the waveform generators. This resulted in a minimal bus load, since only a few data words were needed to specify the required warning message and its status from the waveform generator to the communications audio management unit.

- In examining locations for the bus control, executive control, display management and symbol generation functions, there were several possible choices. The most obvious way was to combine the bus control and executive control functions in a single equipment and the display management and symbol generator functions in a separate equipment. However from analysis of databus traffic it was shown that the databus traffic could be significantly reduced by combining all four functions into a single equipment. It was also shown that by using a single equipment significant savings in weight, volume, cabling, power and cooling would result and finally it was established with potential suppliers that this solution was viable. Thus a specification for a waveform generator which undertook all four functions was put out to tender.

  In this specific case the main operational software which included the bus control transaction tables, executive control, warnings and display management functions were produced by BAe while other software functions for bus control algorithms, symbol generation, built-in test and basic system operating modules were produced by the equipment supplier. This combination of software within individual processors was satisfactorily achieved through the use of common software standards and tools.

It must be recognised that the system design processes described here are neither simple not has it been possible to cover all the aspects in this paper. Considerable design effort was provided in parallel,by equipment engineers and other disciplines. However the basic planning, the design steps and the use of the formal design method CORE all resulted in an unambiguous set of system requirements which would not have been produced by conventional techniques. These enabled the design to be successfully implemented.

6. SYSTEM MANAGEMENT AND CONTROL

The structured design process with its pre-defined life cycle stages provided the basis for the management and control strategy. Each life cycle stage had a specified start point, purpose and resultant output. Thus as the life cycle unfolded the successful completion of these outputs provided management with the necessary measure of progress and achievement.

The management control procedures, to which all life cycle stage products were subjected can be summarised as follows:-

- Review
- Configuration Control
- Change Control
- Configuration Status Accounting

6.1 Review

The aim of the review process was to ensure:-

- a satisfactory completion of one stage of the life cycle before commencing the next.
- the correct planning for successful implementation of subsequent stages.

The review itself comprised a technical review and a separate management review. The former checked for technical accuracy, compliance with previous stage and conformance to standards. Since the majority of the lifecycle products were generated in machine readable form, automated checking procedures were used to assist in validating compliance with previous higher level stages and proving technical accuracy.

The technical review was a formal process and all remedial actions were recorded in the form of a review report. In addition an independant quality control report was also produced. All such products from the technical review and the reviewed item were passed onto the management review. This latter review ensured that all required actions identified and recorded at the technical review, were implemented in an accuracte and timely manner and that adequate resources were available to implement the following stages.

6.2 Configuration Control

Following a successful review the items were subjected to formal configuration control. At this point the item ceased to be the property of the author and became a project item. All formal distribution of the system life cycle products was performed via configuration control. Thus all recipients knew that the work could proceed against a formally established and controlled baseline. Any subsequent change to such baselines were therefore communicated to all affected parties in a controlled authorised manner. As a consequence the work of each life cycle stage was initiated against configuration controlled baselines from the previous stages.

Configuration control was imposed from the first product of the first stage of the life cycle and maintained throughout its entirety. Obviously as the life cycle unfolded and more detailed design work was pursued, previous work was exposed to close scrutiny and errors were detected.

To accommodate such design iterations, for whatever reason, an effective change control procedure was rigorously maintained as part of the configuration control process.

6.3 Change Control

All configuration controlled items were subjected to a rigorous change control procedure. This is particularly important in an 'integrated system' since a single change in one area can cause changes to several processing elements elsewhere in the system. Thus before approval was granted, on any change, all potentially affected areas were canvassed for their comments on the change implication including effects on system design, timescales, manpower effort and costs etc. All aspects of the change process therefore involved the project as a whole rather than individual areas and for this reason the change process was managed by and co-ordinated by an independent configuration control group. This ensured that the correct procedures were enforced and that all relevant data was obtained so as to make the necessary valued judgement as to whether the proposed change should be authorised for implementation. In certain cases formal reviews were held if the change was in any way controversial or extensive. No changes, however small, were allowed to be implemented without formal approval via the agreed procedures.

In addition to managing the change process, the configuration control group also monitored and assessed the change process. Each change was uniquely identified and categorised in order to formulate change statistics. These records provided a high level of visibility to management into how the project was progressing. On EAP it was clearly evident that the adopted design approach had proved invaluable in identifying the majority of errors in the early design stages where they were most easily and cheaply corrected, i.e. in the design paperwork rather than in the defined product. This experience suggests that management should expect to see a large amount of change in progress throughout these early life ycle stages which only decreases as the systems testing stages are reached. This volume of change during the design process, whilst desirable, poses significant problems which can only be contained with the strict aherence to configuration control procedures and the maintenance of configuration status accounting.

6.4 Configuration Status Accounting

In a project with a large number of configuration controlled items and an even larger number of changes in progress, the status accounting of these items becomes essential. All project participants must be kept informed on a frequent and regular basis about the current status of all configuration items, i.e. their issue number and all applicable authorised changes to that issue. This process is merely the communication of current design data baselines to all parties involved in the furtherance of the systems design in accordance with those baselines. A configuration status report also communicates progress and achievement to project management. This report covered

- overall progress
- a list of all configuration items, their issue and change status
- all new items subjected to configuration control
- all new issues of items indicating embodied changes
- all new changes including their status
- change analysis data.

7. OBSERVATIONS AND CONCLUDING REMARKS

The use of the structured design approach together with its associated tools undoubtedly contributed to the success of the EAP in designing the avionics system to a high standard in an extremely short timescale. The following specific points are considered to be worthy of note in summarising what was learnt from this exercise and in providing pointers to the design of future weapon systems.

- The structured approach together with the application of the rigorous management and control procedures enabled realistic programme plans to be produced and provided a high level of visibility to management in terms of progress of the design activities. The change statistics proved to be a valuable indicator of how the project expectations and requirements were being fulfilled.

- The undertaking of a freeze of the system requirements prior to starting the functional analysis process and then the application of a strict configuration control procedure which virtually eliminated the introduction of changes to these requirements once the freeze had taken place were considered to be major factors in enabling the programme timescale to be met.

- The use of the structured approach brings about a significant increase in the amount of design documentation produced, however this is greatly assisted by the use of computer aided tools which reduce the labour intensive nature of this task. This increase in design documentation is a significant step forward in overcoming past deficiencies of having insufficient information readily available. It also significantly reduces such tasks as the production of test specifications, customer manuals etc.

- The key to the production of a high quality system design was undoubtedly the insistence on the adherence to the very rigid control and management of the design process and the use of the automated validation tools. CORE proved to be a very powerful tool not only for design but also for fault finding due partly to the extensive documentation. This was found to be very versatile in assisting the engineers to rapidly locate the problem area and correct it. It also enabled changes in the requirements to be introduced easily and rapidly.

- The structured approach to the total system design places more responsibility on the weapon system contractor who carries out the partitioning process, and therefore determines where and how the various functions will be carried out. How much of the resulting activity is undertaken by the weapon system contractor or the avionic equipment suppliers is a matter of debate. It is considered that avionics suppliers will continue to implement the specialist functions such as sensors, displays and processor hardware but the definition of the on-board software will become the responsibility of the weapon system contractor.

- As well as the tools associated with SAFRA which were used to assist in the design of EAP systems, use was also made of mainframe text processors, minicomputer word processors, standard proforma and data base tools. For future projects, the extension to a comprehensive, centralised, computerised engineering database is considered to be highly desirable.

- In any project there is a need for effective configuration control throughout the design phase. On EAP this was handled by manual means supported wherever possible with computer aids. As the size of the project increases and the use of a centralised engineering data base with multi-user access is established, so automated configuration control tools must be available.

- It is considered that complex system requirements cannot be accurately described in plain 'english' text. The use of tools such as CORE generate their own design language and introduce a need for training not only of the system design engineers but of all the personnel who will be associated with the project such as test engineers, support engineers and in particular managers and representatives of the Customer.

- A major difference between the structured 'top-down' approach compared to existing 'bottom up' approaches, is the elapsed time before some functioning of the system can be seen. In the latter case it is usual for some areas of the system to become visible early in the programme, but in the former case the system tends to come together all at once, albeit on time.

In conclusion it must be admitted that there were doubts during the initial stages of the project as to whether the structured design approach was sufficiently developed to enable us to achieve our declared objectives. In retrospect the success of the project in achieving 35 flights in its first 30 days with no requirements for system changes, shows that the doubts were unfounded. In particular, the success of the structured design approach applied to the avionics system in terms of the quality of design, the timescale achieved and its supportability were beyond our expectations and indicates the way forward for the design of the more complex weapon systems of the future.

ACKNOWLEDGEMENTS

The authors wish to express their thanks to the members of the EAP System Teams at BAe Warton and colleagues from the various Avionic Companies in Great Britain, Germany and Italy, who undertook their own part of the total systems design which ensured the success of the EAP. The Directors of British Aerospace are acknowledged for their permission to produce and publish this paper. The views expressed, however, are those of the authors.

This work has been carried out with the support of the Procurement Executive, Ministry of Defence.

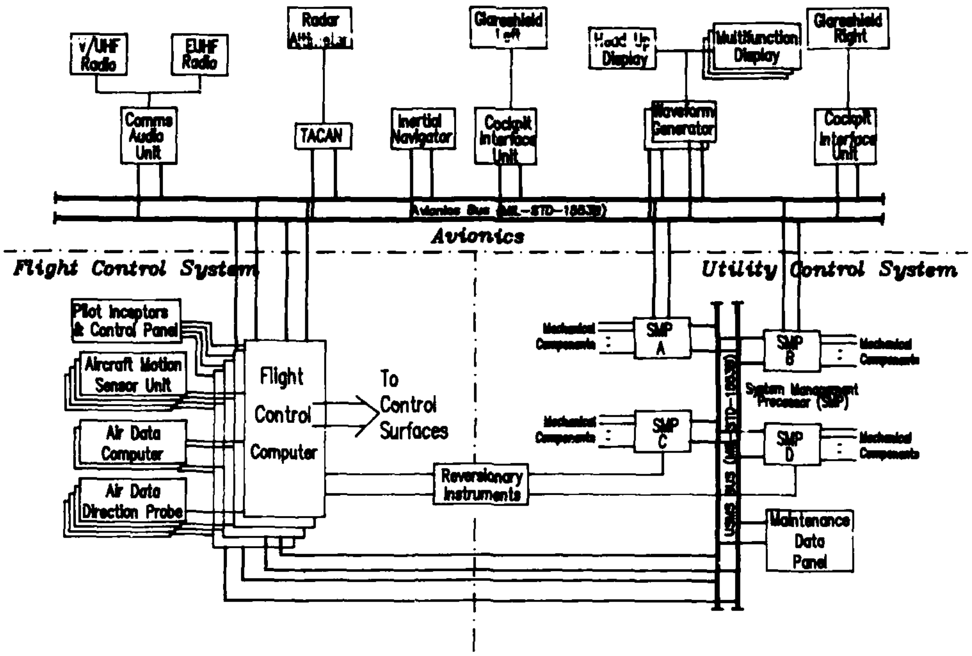


Figure 1. EAP Systems
(Simplified Architecture)

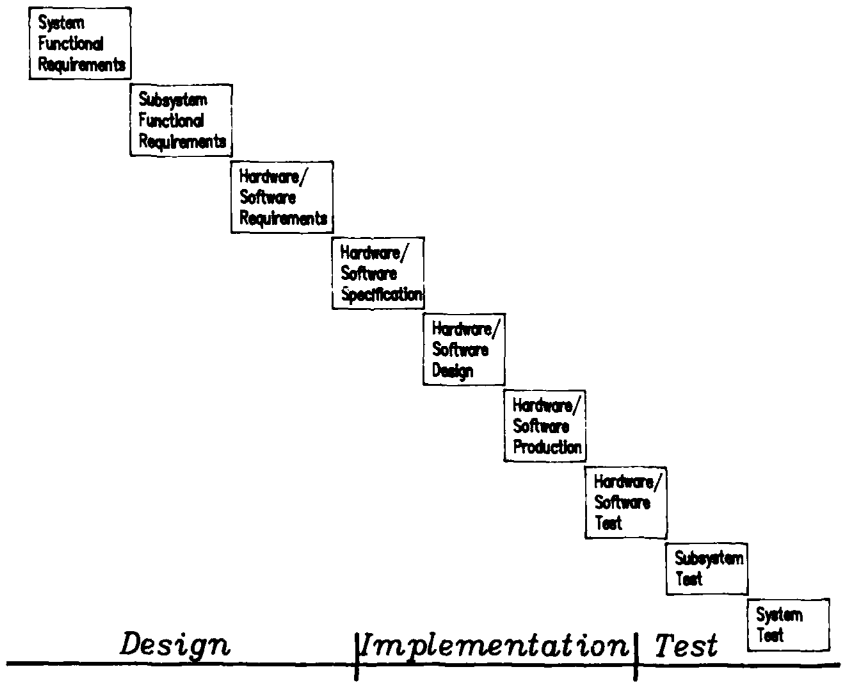


Figure 2. System Life Cycle

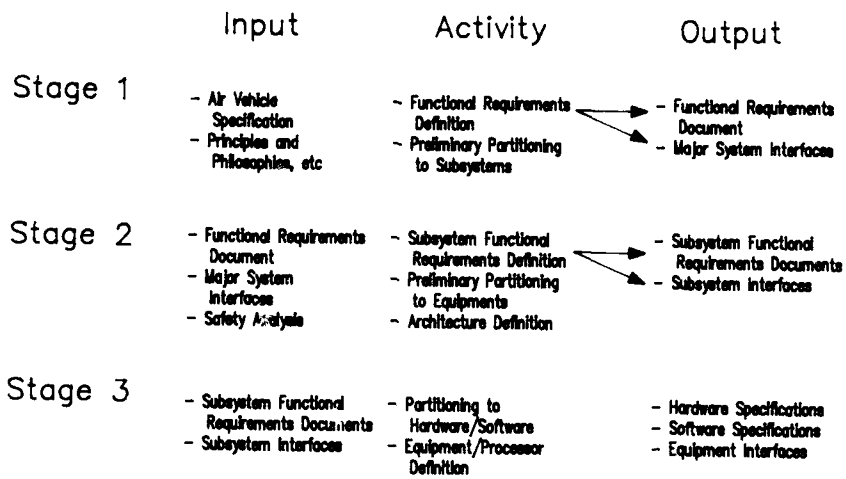


Figure 3. System Design Route Map

DISCUSSION

P.Simons, US

Do the software tools for this process support multiple decompositions of a functional analysis?

Author's Reply

Figure 3 of the paper shows that hardware assignment or design occurs at the end of stage 3 of system design, after the global system and subsystem functions have been identified. System design is driven by mission and phase of flight requirements. Decomposition at stage 2 (see Figure 3 of paper) identifies subsystem requirements at the attack, flight control, display and control, etc., levels. This includes all mission requirements. Particular mission analysis can be performed at this point, including the system's offensive and defensive abilities. The system's defensive abilities are generally compromised in the UK because of the requirement to use existing equipment provided to the aircraft constructor by the government.

G.Bouche, GE

Do you know about national or NATO activities aimed at standardization of system design tools such as CORE?

Author's Reply

We are not aware of any national or NATO activities to standardize system design tools. Within British Aerospace, there is an intent to standardize on CORE. For EFA, the customer requires CORE/EPOS to be used throughout the design. This requirement is driving CORE/EPOS as a common project (EFA) design tool.

DEVELOPMENT OF A GENERIC ARCHITECTURE

by
Christina Berggren
International Business Machines Corporation
Federal Systems Division
MD 0906, Owego, NY 13827-1298 USA

SUMMARY

A new generation systems architecture being developed at IBM in Owego, NY, is designed to bridge the gap between today's 1553-based systems and the fault-tolerant, totally integrated systems of tomorrow. This paper describes a novel approach to system functional area partitioning and the design of this generic distributed real-time architecture. The architecture incorporates new military standards in development

INTRODUCTION

Next-generation avionic systems cannot be developed by traditional means. The simulation of a centralized architecture system and the gradual replacement of the simulator with the prime system 'black boxes' during development is a discipline well suited for today's technology-based systems. The architecture of test beds supporting the development of future avionic systems, however, must map onto the architecture of the system being developed, both hardware and software. A distributed fault-tolerant system with parallel execution and dynamic task allocation to processors cannot be simulated by a centralized sequential architecture.

The architecture of the Advanced Systems Development Laboratory (ASDL) at IBM Owego is designed to emulate systems of the future on commercial hardware. Validated system designs can then be hardened with little risk.

The ASDL, presently in concept validation phase, employs a fully distributed, data driven architecture. It is a hierarchically controlled parallel pipeline of alternating layers of self-contained Ada* tasks and a communication architecture based on next-generation military standards.

The decomposition of this next-generation system required innovative approaches and a new way of thinking when the requirements were partitioned into functional areas of manageable subparts.

FUNCTIONAL AREA PARTITIONING

The objectives used to guide partitioning of the ASDL into functional areas were reusability, flexibility, and extendibility. Application-unique requirements were isolated by extracting generic services applicable to all systems, and this resulted in an architecture partitioned into four functional areas as shown in Figure 1.

The application-specific requirements were allocated to the Simulation Models Functional Area and Data Translation Functional Area and the generic system services to the Communications Functional Area and the Execution Control Functional Area.

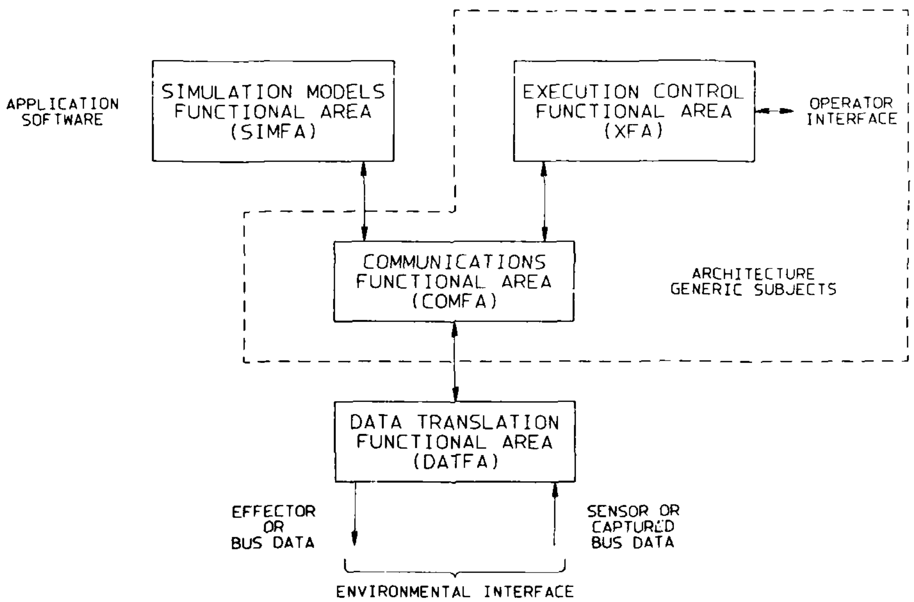


Figure 1. ASDL Functional Areas

*Registered Trademark of United States Department of Defense

Traditionally, system-level functional area partitioning maps onto hardware. In the ASDL, system partitioning transcends conventional hardware and software boundaries. Figure 2 shows the ASDL mapping of functional areas onto one hardware node in the distributed architecture. Application modules and part of execution control are at the top, and layers of the communications architecture are below.

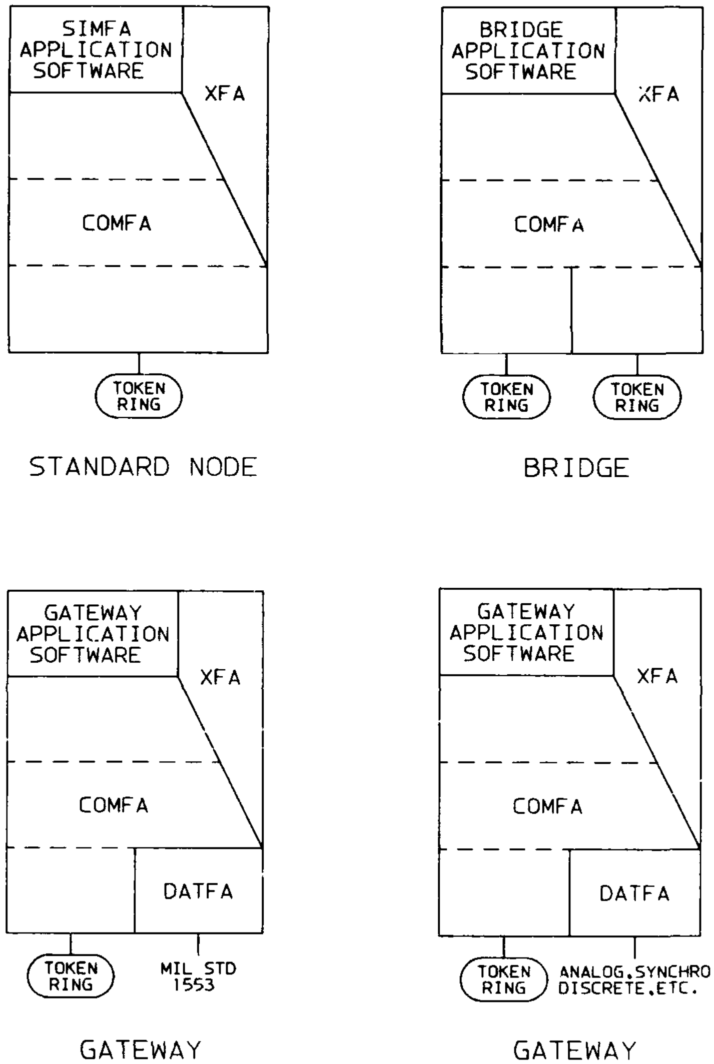


Figure 2. ASDL Node Functional Area Mapping

SIMULATION MODELS FUNCTIONAL AREA

Mission-unique functions have been allocated to the Simulation Models Functional Area (SIMFA). SIMFA is a collection of application-specific Ada modules, each of which accomplish one self-contained task. The software interface to these modules is standardized to allow any number of modules to be merged together. Combinations of modules emulate entities ranging in size from small subsystems to large complex systems of the future.

Intermodule communication at this level in the architecture is logically independent of hardware allocation and supported by the underlying system services. Task execution is externally initiated.

Thus, through system-level partitioning, application- or mission-specific functions were isolated. System adaptation or changes are limited to adding deleting or upgrading SIMFA modules for software.

DATA TRANSLATION FUNCTIONAL AREA

Application-unique hardware interfaces have been allocated to the Data Translation Functional Area (DATFA). The DATFA translates the standard intra-ASDL digital format to/from analog, synchro, discrete, or whatever the specific need may be. DATFA also converts data to formats required for specific data buses. The first ASDL DATFA implementation is a token passing ring to 1553 gateway. This gateway will allow for hybrid systems during the transition period into next-generation military architectures.

The main ASDL development efforts, however, are focused on the generic subsets of the architecture: the Communications Functional Area and Execution Control Functional Area.

COMMUNICATIONS FUNCTIONAL AREA

The Open Systems Interconnnection Reference Model of the International Standard Organization (ISO) was evaluated for applicability to the development of the ASDL communications architecture. Its seven-layer model defines the partitioning of the communication services between heterogeneous end users in an open network type system.

The types of communication services required for a real-time distributed system differ significantly from those of an OSI application non-real-time system as do the requirements for the implementations. A large percentage of the data in real-time applications is sensor data sampled periodically as often as sixty times per second.

To establish virtual circuits beforehand and retransmit erroneous data for reliability is neither needed nor desired if new data will be available in 16 milliseconds. The processing overhead of the communications protocol may require more time than that.

The requirement for a high-speed implementation of a completely connectionless, unacknowledged datagram service is one area where military distributed real-time systems differ from traditional systems. Naturally, military systems also require the reliability of a connection-oriented service with application-level acknowledge for mission implementation and system execution control.

The implementation of these services has different design drivers. For real-time systems, as the name implies, reducing the time required for a task to communicate with another is a primary design driver, thus minimizing the end-to-end delay through the system.

The implementation of the ASDL communciations architecture, however, does take advantage of the OSI philosophy and structure. Partitioning the services into layers of separate entities provides flexibility that allows change. Changing or adding to a function in one layer will not ripple through the system. The impact of the change will be confined to the one entity being changed.

In order to implement efficient real-time communication services, the communication protocols of the ASDL architecture have been partitioned into three layers:

- Lower Layer (I/O Driver)
- Middle Layer (Communications Control)
- Upper Layer (Application Communications).

LOWER LAYER

The lower layer is the I/O Driver, and it includes the I/O device or LAN (Local Area Network) hardware. The LAN technology chosen is a high-speed token passing ring. It allows for distributed control, deterministic access, and delays that are quantifiable and minimum between geographically dispersed nodes.

In the prototyping facility, the LAN used is an 80Mb/s token passing ring manufactured by Proteon Inc. and employs a round robin type access method. The lower layer and the interface to it were designed with provisions which will make it easy to replace the layer when hardware becomes available for the next-generation military standard token passing ring. (This new standard, defined for fault-tolerant distributed real-time systems, was developed by the SAE-AE9B* subcommittee and is presently going through the approval cycle.)

The I/O device hardware contains high-speed input and output buffers and a state-machine that controls the interface to the communications media. The host computer is interfaced via the CPU internal bus.

The primary function performed by the lower layer software is to move data between the host's internal memory and the input/output buffers. There is little intelligence in this layer; it does not determine memory address or type data move. It reports errors or anomalies in the lower layer or the transmission media to upper layers for action, but it does no error handling.

MIDDLE LAYER

The middle layer, Communications Control (Com Cntl), controls all node external communications.

For incoming data, Com Cntl analyzes the header of the packet before data is moved from the input buffer on the I/O device. Based on the data's logical source, Com Cntl defines memory destination for the data. The type of operation for the I/O Driver to perform is defined in the header Message Category word and initialization tables. This can be queue algorithm, using DMA, programmed I/O, etc., to move the data to another I/O device for gateway or bridge functions, or Com Cntl can direct the I/O device to discard the data if it is no longer required.

For outgoing data, Com Cntl performs the logical source to physical destination address translation and prepares the packet header before directing the I/O device to transmit the data. This address translation is a self-contained entity that can easily be removed when the MIL-STD token ring is integrated, since it supports logical addressing in the hardware.

In order to reduce the end-to-end delay, data is only moved once as it passes through the communciations architecture, and that move is between host memory and the high-speed input/output buffers in the I/O device. Buffer pool techniques are used to sequentially pass control of access to the data from the application onto the upper layer, to the middle layer, and to the lower layer of the communications architecture.

---

*Society of Automotive Engineers -- Aerospace Equipment

UPPER LAYER

The upper layer, Application Communication (App Com), controls all communications internal to a node.

Logical addressing is used throughout the system such that a task never knows where another task that it communicates with is located physically. App Com thus has the responsibility to define whether data generated within a node is required by another task within the same node, or if control of the data is to be passed to Com Cntl for subsequent transmission to another physical node

All task communication management has been allocated to App Com. Each task resident at a node has a corresponding communication management task in this layer. This App Com task is responsible for receiving and keeping track of all data required by an application for execution. For instance, if an application requires data from one internal source and two external sources, App Com will take access control of the data as it arrives, and when all necessary data is received, access control will be passed and the application task posted for execution.

As mentioned earlier, military real-time systems require two classes of communication services: a connnectionless datagram type service, and an application-level acknowledged message. In the ASDL architecture, the application-level acknowledge is supported in App Com. App Com's communications management includes waiting for acknowledgement to transmission of these types of messages for a predetermined period of time, retry of message transmission, and alerting system services if no response is received. App Com also provides system services with logically named error messages from the other layers for initiation of programmable, *user-defined error* handling/reporting tasks.

LAYER INTERFACE

Key to the implementation of the ASDL architecture is the design of the layer interface. The structure of the interface has been standardized to allow near indefinite flexibility and extendibility. Functions may be added or deleted from a layer without affecting other entities. Actually, the structure and implementation of the layer interface permits entire layers to be added, if required, for additional non-real-time services.

The layer interface is designed utilizing the OSI concept of Service Access Points (SAP). SAPs are memory pointers shared by entities which communicate with each other across a layer interface. Communications via the SAPs occur in the following manner. An entity wishing to operate on an entity in an adjacent layer prepares a control word defining operation type and parameters required for the operation. A memory pointer to the control word is then loaded into the specific SAP, and CPU control is passed over the layer boundary. The receiving entity reads the SAP, finds the pointer to the control word, and then decodes the control word to perform the desired operation.

Each layer boundary transition costs CPU time and adds to the end-to-end delay of the system. In the ASDL architecture prototype, the design of the layer interface was focused on speed and resulted in a fast, efficient, and flexible implementation

In the data-driven system architecture prototype, there are two SAPs between each I O Driver and Com Cntl and two SAPs between Com Cntl and App Com, one for each direction of operation, IN or OUT. Figure 3 depicts the layer interfaces with associated SAPs and operation direction. Communications occur in both directions over a SAP. Any one operation may require a sequence of control words and responses to be communicated via the SAP.

Direction of operation is defined by who operates on whom; that is, who initiates the operation. For the implementation of the data-driven architecture, this means that for any given message arriving at a node via the token passing ring, first the I O Driver will operate on Com Cntl, then Com Cntl operates on App Com, which in turn may operate on the application task. For an out operation, the sequence of operations is reversed except for the application task.

By definition, only control tasks can operate on another entity; mission-specific tasks get operated on and respond.

The ASDL interface design allows flexibility in allocation of communications architecture between main processor and I O processor. Communication-intensive applications may require a separate I O processor within the node, but for low to normal communication needs, this processing, or any layer of it, can easily be moved to the main processor.

EXECUTION CONTROL FUNCTIONAL AREA

In the ASDL implementation, SAPs are not only nuclei of communications but also of control. Any operation can, at the interface, be halted, resumed, or, for debug purposes, single stepped. This allows a total integration of operation management and control functions with the communications.

Operation management and control is one of the system tasks allocated to the Execution Control Functional Area (XFA) of the ASDL architecture. The design supports standard control access points via the SAP implementation such that all or any of the layers of the architecture or application tasks can individually be executed under XFA control algorithms. These algorithms can be dynamically changed, depending on system state. A straight mission priority data-driven task scheduling algorithm, for instance, can be replaced with a best effort decision-based algorithm for temporarily degraded system states during fault recovery when only partial data sets may be available for processing.

This task scheduling algorithm decision task is a subset of the overall resource management task resident in XFA. Other tightly integrated subsets include redundancy management and anomaly reaction.

XFA contains a hierarchical control structure with some subsets resident at all nodes all the time. These subsets include local node states; that is, dynamically updated resource status data bases defining tasks allocated to said nodes and their communication needs and execution state, as well as any locally detected anomaly condition.

XFA tasks on the next level of the hierarchy can be resident at any node but need only be resident at one (or more dependent on level of redundancy) at any given time. The resource allocator task, for instance, queries other nodes for processing load by reading the local resource status data bases via a roll call before making any task configuration changes.

The system anomaly handler task is also resident in this next higher level of the XFA hierarchy. A combination of periodic health checks from the individual nodes and anomaly reports from the local XFA anomaly handlers provides for automatic reconfiguration in the event of failures.

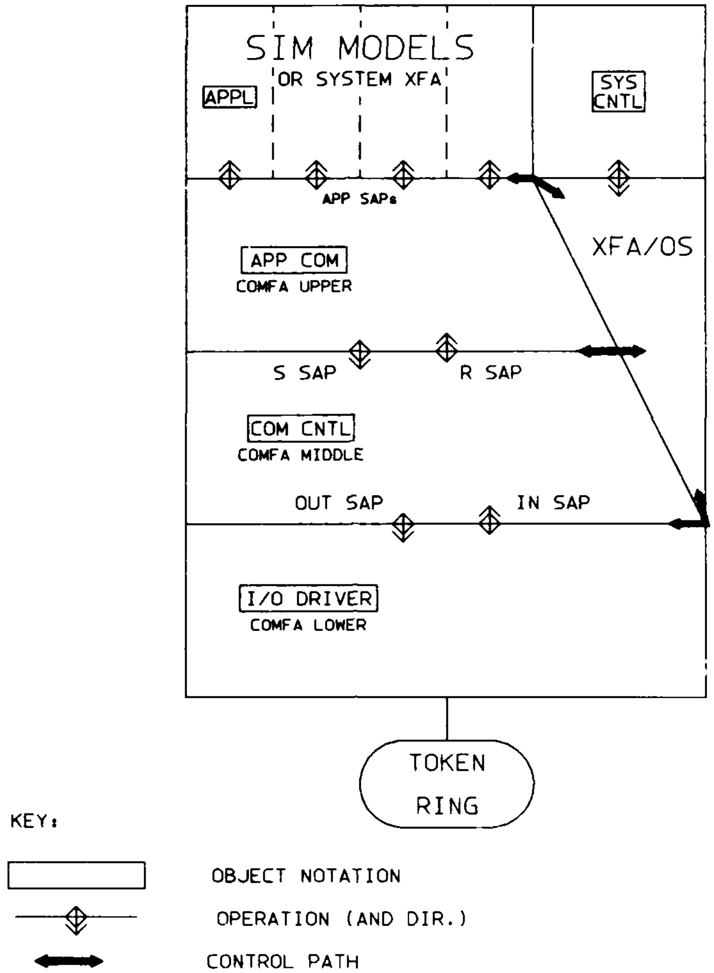


Figure 3. ASDL Architecture: Service Access Points and Control Paths

The extent of system automation is operator defined. XFA is also where the user interfaces with the system. Here, the task partitioning between man and machine can be dynamically changed at any point, depending on such priorities as mission, safety, or security.

The user defines what events he/she would like reported in real time for operator intervention and what events can be handled automatically by the system. These user-defined programmable instructions are tightly integrated in the hierarchical control structure. XFA resident control states, in conjunction with the SAPs in the communications architecture, gate system commands to be executed and prevent nonauthorized recention of messages.

Data bases and their management are also allocated to XFA. This includes both the distributed dynamically updated fused sensor state and resource status data bases and the main library of system available functions. A menu-driven interface allows user friendly access.

CONCLUSION

The ASDL architecture prototyping facility consists of commercial minis and super-minis connected via a fiber optic, high-speed token passing ring, as well as special purpose AI processors and other arithmetic units, tightly integrated with a high fidelity, one-man cockpit and out-the-window displays.

Early development focused primarily on the communications architecture and the implementation of a high-speed layer interface. During the next year, prototyping of the system design will continue with XFA for a total validation of the concepts, and optimization of the integration of system controls with the communications.

The ASDL architecture in development at IBM Owego will allow evaluation of advanced avionic architectural concepts and facilitate the phased implementation of next-generation systems.

TEST PHILOSOPHY OF THE EH101 INTEGRATED AVIONIC

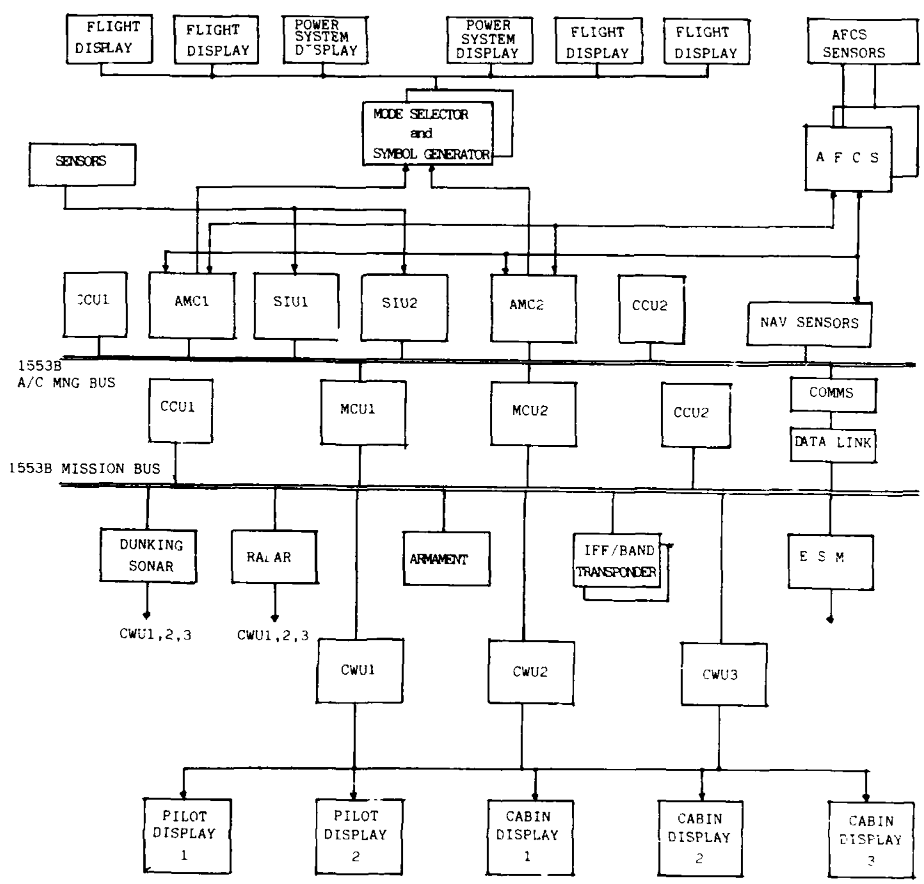
by
E. Galli
AGUSTA SISTEMI S.p.A.
Via Isonzo, 33
21049 Tradate
VARESE - IT''Y

SUMMARY

The intent of this paper is to succintly outline the philosophy employed by Agusta during the development and testing of the EH101 integrated avionic naval helicopter. The paper is written following the building blocks of the avionic under test (see figure below on which the avionic architecture is shown).

1. SYSTEM DESCRIPTION

The EH101 avionic system is divided into two main processing areas: the Aircraft Management System (AMS), and the Mission Avionic System (MAS).



The Aircraft Management System consists of:

- Two redundant Aircraft Management Computers (AMC): i.e., one active, the other in back-up mode.
- Two Sensor Interface Units (SIU) to handle all the analogue and discret signals coming to/from the helicopter's sensory system.
- A Control Panel (CP) that allows the pilot to select operation (automatic or manual) of the system.
- A Data Transfer Device (DTD) to do the download/upload exchange from external equipment such as preflight data and maintenance information.
- Two Common Control Units (CCU) which permit the pilot and co-pilot to interact with the system (AMS and MAS).
- Symbol Generators (SG) manage the Electronic Instrument System (EIS) programed with the formats required for the display of navigation, flight and power systems.
- A Navigation Pack which includes Air Data Unit (ADU), Radar Altimeter (RA), Doppler velocity, Tacan, Global Positioning System (GPS) and Inertial Reference Units (IRU).

The AMS performs management of all the basic functions of the helicopter. These basic functions include Navigation, Maintenance, Performance computation, Communication, and the monitoring of the helicopter systems (i.e., rotor, transmission, engine, fuel, electrical, hydraulics, etc.).
The Mission Avionic System is based on:

- Two redundant Mission Computer Units (MCU), same concept as above: active and back-up.
- Two Common Control Units (CCU) that allow the cabin operators to interact with the system (AMS and MAS).
- Common Waveform Units (CWU) to manage cabin and pilot tactical situation displays and tabular information.

Sensors dedicated to perform the Antisubmarine Warfare (ASW), Antisurface Vessels (ASV) and Electronic Warfare (EW) are subdivided into:

- Underwater Sensors
  - Sonar subsystem
  - Magnetic anomaly subsystems
- Surface Sensors
  - Radar subsystem
  - Electronic Support Measure (ESM)
- Identification Systems
  - Interrogator Friend Foe (IFF) Transponder
  - Interrogator Friend Foe (IFF) Interrogator
  - Intermediate Band Transponder
- Communication System
  - Data Link subsystem
- Armament System
  - Weapons management subsystem
  - Store management subsystem

The two MCU perform all the processing required for the mission operation, collecting, processing and storing tactical information coming from the Mission Systems. Each of the two areas (AMS and MAS) use as the main data transmission medium a dual redundant MIL-STD-1553B data bus, the AMS also uses ARINC 429 lines to communicate with most of the navigation sensors and the Automatic Flight Control System (AFCS).

2. TEST DESCRIPTION

In order to carry-out the entire integration and testing of the Avionic System four main phases have been identified.

DEVELOPMENT

In this phase each subsystem or piece of equipment is designed and developed in accordance with the avionic system requirements.

TEST PHASE AT SUBSYSTEM OR EQUIPMENT LEVEL

The assessment with regard to the requirements is carried out.

FIRST INTEGRATION

At this level integration of the Aircraft Management System is conducted parallel to the integration of the Mission Avionic items. Testing of these items is contingent upon computerized structures called Rigs. The equipment under test is controlled and stimulated by these Rigs. The basic configuration of the rigs is a host computer, a set of 1553B and ARINC 429 bus stations, subsystem simulators, input/output stimulators and control and display units.

FINAL OVERALL AVIONIC INTEGRATION

At this point, the Aircraft Management System and the Mission Avionic System are fused, tested, and validated. Again such testing is made feasible by means of the Rigs. The above mentioned phases are conducted to achieve the following goals:

- Minimize the number of prototype flight trials necessary to test the integrated avionics.
- Reduce the technical risks associated with software/hardware integration.
- Accelerate the iterative loop (i.e., test/design changes/retest) using suitable test programs and resulting data analysis.
- Deploy an efficient test battery to execute all test runs and cases required to qualify and certify the systems.
- Facilitate the integration of new subsystems, affording timely responses to experimental software and/or hardware modifications. This feature transcends the design and development phase. It will be active throughout the helicopter's life cycle.

Let us explore each of these phases in more detail. The Aircraft Management System avionic test administers four separate trials. Trials occur at each of the following levels:

Aircraft Management Computer Test

The aim of this test is the validation of the hardware architecture when put together with the basic software under more severe ground loading conditions than those encountered during flight. Testing focuses on the following main areas of interest:

- Interface of 1553B's input/output
- Arbitration of shared random access memory
- Degradation of performance under multiprocessor interference and interactive conditions
- Redundancy management.

Tools and techniques used to detect errors during testing are:

- Real time simulation
- Failure mode generation in both software and hardware items
- Failure analysis
- Artificial stress of C.P.U. capabilities
- Special test software used to establish the readiness of basic software and hardware prior phase activity.

The A.M.C. rig is composed of 1553B & ARINC 429 bus stations. Both bus stations are remotely controlled by the host computer which provides them the necessary data to simulate equipment not yet available. It also monitors and analyzes output of the AMC on the 1553B bus and ARINC 429 lines.

Sensor Interface Units Test

The purpose of this test phase is to verify the proper operation of the SIU integration with actual sensors. The SIU are programed with a select input/output polling sequence; the sequence having been performance verified. The input/output data exchange with the 1553B system can also be defined, programed and verified. Testing focuses on:

- 1553B Input/Output interface
- Stress incured when CPU respond to different input/output sequences and sampling rates
- Real time simulation
- Failure mode generation in software/hardware items
- Failure analysis
- Assessing results, CPU loading, unit size and timing.

The SIU is also tested by the rig. This rig is composed in part by 1553B bus stations which emulate the bus controller commands, monitor and analyze the SIU's data transactions on the bus. The other component of the rig is the emulator of the helicopter sensors (i.e., temperatures, pressure, loads, discrets, frequency, etc.). Such items can also be governed remotely by the host computer.

Aircraft Management System Test

The aim of this test phase is to verify correct integration of the AMC & SIU (the function of each has already been tested). Major tests performed are:

- Start-up functions
- Redundancy management (master/slave) in case of failure
- Cross channel checking function
- Maintenance data bases exchange
- Bus monitor functions.

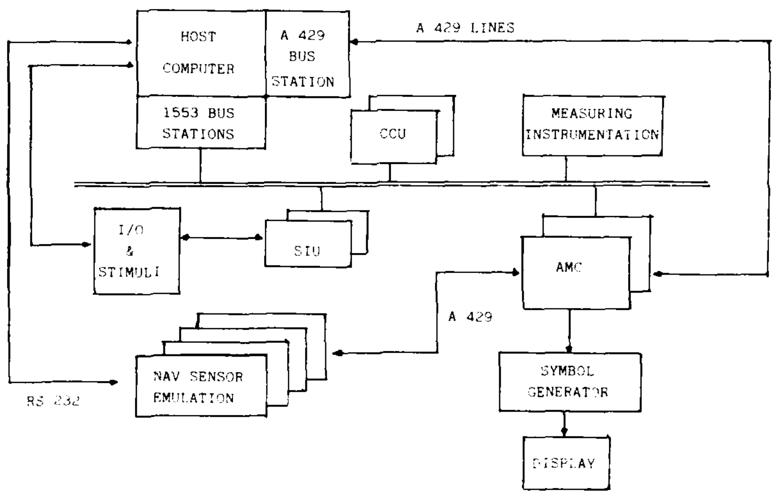
The AMS rig is an amalgamation of the AMC & SIU rigs. It consists of the host computer, bus stations (1553B & ARINC 429), SIU, emulators (helicopter sensors & navigation subsystems), symbol generator, CRT display and common control unit. It becomes prudent at this time to discuss the navigation emulators. They are comprised by a set of single board computers linked to the host computer through an asyncronous line. Each single board computer emulates a specific navigation subsystems in terms of timing, protocol, failure & signal characterization. The host computer furnishes on-going, updated data derived from the flight plan scenario.

Aircraft Management Test

The activities examined during this assessment are:

- Hardware/Software interfaces, functions & capabilities
- System performances
- Verification of reversionary modes
- Collection of reliability data
- Optimization of the download/upload procedure
- Support of flight planning
- Check system validity before flight test.

At this point the aircraft management system rig gives priority to monitoring, acquiring, and analyzing data. Emulation activities previously provided by the rig are no longer required as the real equipment is now installed. A block description of the rig is given (see the following) on which all the elements constituting it appear with their relevant interconnections.



The foregoing described the A/C Management System elements of the integrated avionic. Let us turn our attention now to the Mission Avionic System. The configuration under test in this portion of the project, like its counterpart the AMS, follows a development concept of ever increasing complexity. Dedicated test facilities are provided (special to-type test equipment) for evaluation of equipment requiring particular attention (i.e., sonar, radar, armament, ESM). The rig is also constituted of Host computer, 1553B Bus stations and a common wave-form generator unit. The rig is tasked with providing the activities listed below:

- Hardware/software interface integration
- System performances (bus occupancy, CPU load, data security, etc.)
- Verification of reversionary modes
- Collect reliability & maintainability data
- Optimize download/upload procedure
- Support software validation
- Verify build-in test functions
- Support flight planning.

Note, the functions as outlined in the preceding do not include data handling management of the tactical scenario as needed by the operators. This job is performed on what is known as the Mission Software Development Rig (MSDR). All aspects of operator interfaces, data routing, and the human factor are studied on the MSDR. Off-the-shelf equipment (computer, graphic generators, CRT, compilers, data base, etc.) are used. This avails the flexibility necessary for a ready analysis of an existing set-up and its editing. This mode of operation is undertaken to prevent waste of manpower during the on-board software development, validation & testing phases.
The development of the two systems, Aircraft Management and Mission, is effected in two distinct environs. Only marginally taking into consideration is the information interface of the two systems. At this point consolidation of the two systems and their respective rigs takes place. Synchronization with regard to a common scenario, also occurs at this juncture. Having achieved merger the rigs are referred to as the Overall Integration Rig. The principle activities executed at this level are:

- Validation of interfaces between the two areas
- Initialization functions
- Failure modes
- System performances
- Total system validation.

The Overall Integration Rig is subject to scrutiny similar to that of the Aircraft Management and Mission stage. The prime uses of the Overall Integration Rig are to:

- Analyze data captured by the assigned bus stations
- Emulate, in a dynamic situation, any required subsystem functions
- When possible, replay excerpts from actual flights that require further examination.

3. CONCLUSIONS

As illustrated, the testing activity is not incumbered by external delays. Testing can be carried on, as preliminary, by the use of emulation capabilities made available by the rigs. The EH101 Avionic Test Activities are in the early stage, at present, so a deep analysis can not be conducted. However, it can be stated that the Agusta EH101 Integrated Avionic test philosophy has responded well to our needs.

Systems Engineering Technique

Leonard Karas and Donna Rhodes
IBM Corporation, Federal Systems Division
Owego, New York 13827
USA

1.1 (NU) SUMMARY

This paper provides an overview of the Systems Engineering Technique (SET), a methodology developed at IBM Federal Systems Division in Owego, New York. SET has been developed to effect improvement in both quality and productivity aspects in the development of avionics systems. The methodology synthesizes the best features of existing development methodologies into a single core procedure which is equally applicable throughout the systems development phases of complex systems. Six key areas emphasized by SET are discussed, and the concept of systems engineering measurements is introduced as the means to evaluate system quality and productivity. SET is being applied to the development of avionics systems at IBM Owego and has proven to be effective in improving specification quality.

2.1 (NU) INTRODUCTION

Two major aspects of development that must be addressed by a systems engineering methodology are quality and productivity. Quality concerns the effectiveness of the system in meeting the requirements. Productivity concerns the effective use of resources in system development. Tools to manage resources and automate time-consuming or manually impossible tasks contribute to productivity in the development effort. The use of standardized models and procedures is a key element of both quality and productivity; reuseable components is another.

Methodologies which address these issues are by no means in short supply. What does seem to be lacking today is a methodology which addresses all of the key elements influencing system quality and productivity throughout the system development phases. Figure 1 describes the system development phases, the associated program review points and the system engineering specifications developed for each program review.

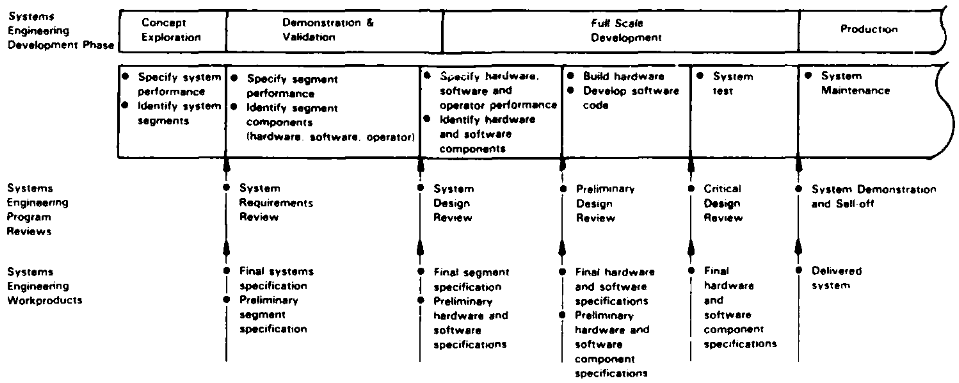


Figure 1. (NU) Systems Engineering Development Phases

SET was developed in an effort to synthesize all of the best methodologies within a practical framework equally applicable throughout each phase of the systems development. As a result, SET places emphasis in six key areas. They are:

- Use of three information models to develop the system to the fullest extent possible within each incremental level
- A combination of top-down and bottom-up approaches which allows for parallel development of requirements and design
- Integration of the engineering disciplines in all aspects of the systems engineering process
- A repetitive procedure that is applied to each incremental level of system definition

- Use of a plan to manage and coordinate each incremental level of system definition
- A tightly controlled exit from each incremental level of system development

These key areas provide for tighter control over the systems development, both within and between incremental levels of system definition. Three of these key areas are addressed within each incremental level and provide a more fully developed system through the use of models and processes which maximize involvement of the engineering disciplines and customer.

3.1 (NU) AN INCREMENTAL LEVEL OF SYSTEM DEFINITION

In each incremental level of system definition, SET provides for:

- Use of three information models to develop the system to the fullest extent possible within each incremental level
- A combination of top-down and bottom-up approaches which allows for parallel development of requirements and design
- Integration of the engineering disciplines in all aspects of the systems engineering process

SET employs three models to develop each incremental level of the system. The three models are the functional model, the physical model and the operational model. The purpose of the functional model is to refine the customer's requirements into functional areas that make the system definition and development manageable. Given a set of system functions, the physical model is used to develop a physical implementation (design) that is feasible. A feasible design is one that can be built within the technology, cost and schedule constraints. A feasible design is also maintainable, reliable and supportable. The purpose of the operational model is to take the functional requirements and the physical design for an incremental level and test them for suitability. A suitable system is one that satisfies the customer performance requirements, is useable, is maintainable and is reliable. Figure 2 provides a summary of the three models.

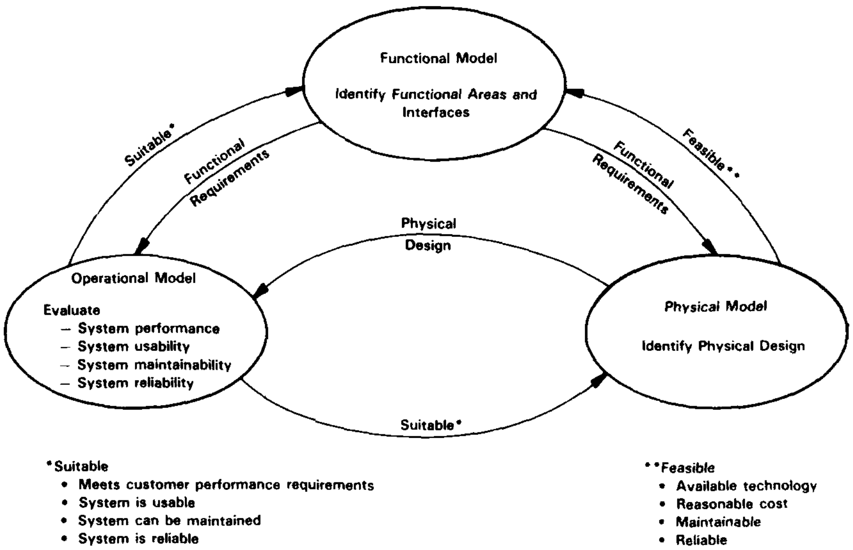


Figure 2. (NU) Three Systems Engineering Models

The development of the three models is an iterative process. Each model has its own unique purpose and overlaps the other two. As each model is developed, it may impact the previous model. Therefore, the development team must cycle between each of the three models until the impacts of one model on the other two are minimized or eliminated.

3.1.1 (NU) THE FUNCTIONAL MODEL (REQUIREMENTS DEVELOPMENT)

The functional model allows the development team to collect requirements into groups so that intellectual control can be maintained over the system development. Each incremental level has its own criteria for developing these functional groups. For example, in the first level of system definition the system segments and their interfaces are identified. In the next incremental level, the segment components (hardware and software) and their interfaces are identified, and so on. An incremental level of the system

cannot be developed without taking a forward look into the lower levels of system detail. In other words, the system must be developed top-down but should be influenced by the bottom-up detail. The functional model allows the development team to look ahead into the lower level system details. That detail is then abstracted into a consistent set of requirements that meet the objectives of a given incremental level of system definition.

As higher level requirements are abstracted from the detail, the systems engineers take advantage of the engineering disciplines and reusability concepts. For example, when developing the functional areas for a software specification, the systems engineers use the guidance of the software development engineers to ensure the functional requirements are suitable for their development of the low level design. In addition, the software development engineers, in conjunction with the systems engineers, will identify and utilize reusable components from similar systems that were developed. The reusable components reduce the cost and the time necessary to develop follow on systems of a similar type.

Each incremental level of the system involves a look ahead from an existing level. Since the requirements from the present level drive the next level of system definition, a traceable link is established from one incremental level to the next. This allows traceability of the lowest level requirements back to the original requirements from the customer. Figure 3 is a typical example of the application of the functional model to an avionics system.

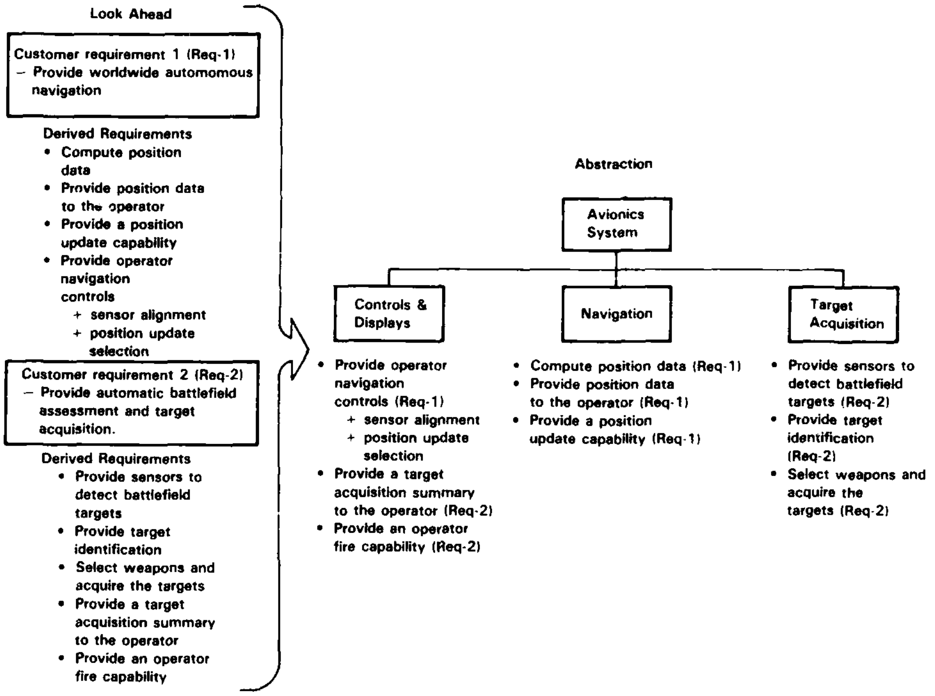


Figure 3. (NU) Avionics System Functions

As indicated in the figure, lower level requirements for the system are developed by analyzing each customer requirement. These lower level requirements are then collected into functional areas using the criteria that is appropriate for the incremental level being developed (e.g., identifying the system segments when developing the system level specification). Each function is shown in Figure 3 with a list of detailed requirements that were used to develop that functional area. The customer requirements that drive each functional area are shown. For example, the Navigation functional area is driven by customer requirement 1, the Target Acquisition function is driven by customer requirement 2 and the Control & Display function is driven by both customer requirements. These functional groups are now considered in the physical model.

3.1.2 (NU) THE PHYSICAL MODEL (DESIGN)

The physical model allows requirements definition and design to proceed in parallel. Therefore, requirements definition is no longer a separate activity from the system design. Instead, the design is considered during the requirements definition without

locking in the design detail too early. The objective of the physical model is to determine if there is a feasible design that can be used to implement the functional requirements. In making this determination the development team must consider technology, cost, maintainability, reliability and availability of the physical components. All of these criteria must be satisfied to make a solution feasible. In considering the alternatives, the services of the various engineering disciplines are employed. System architects, hardware engineers, software engineers, reliability engineers and maintainability engineers are needed to determine feasibility. The feasibility question is asked and answered at each incremental level of system development. When developing the system specification, the team must select the system segments that are feasible for the system under development. If the system segment specifications are being developed the group must select the segment components (hardware, software and operators) that are feasible for each segment. This process of selection then imposes additional constraints on the lower levels of design.

Figure 4 illustrates a design used during the development of an avionics system at IBM in Owego, N.Y. The physical design consists of multiple processors, memory stores and data busses. This design was selected due to the nigh degree of parallel activities the system must perform to meet the overall performance requirements. An important question at this point is how should the functional processing requirement groups (developed with the functional model) be allocated to the physical processors? This question can be answered by considering the functional requirements and physical design in the operational model.

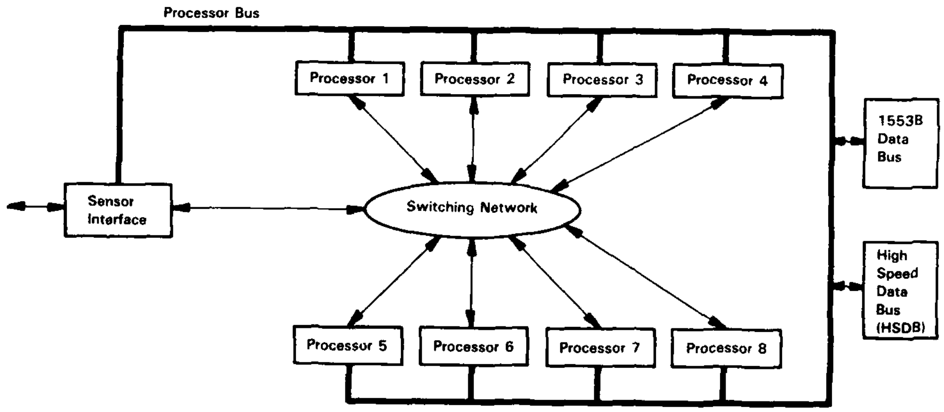


Figure 4. (NU) Physical Design for an Avionics System

3.1.3 (NU) THE OPERATIONAL MODEL

The operational model requires support from a variety of engineering disciplines who provide analysis of system performance, reliability, maintainability and operability. This analysis is used to determine whether the functional requirements and physical design are suitable. *Each incremental level of system definition is deemed suitable if* the functional requirements and physical design form a system that meets performance requirements, is useable by the operators, is maintainable, and is reliable. This is accomplished by mapping the functional areas and the physical components in a time ordered sequence to model the system performance, reliability and maintainability. Operational flow diagrams, derived from a mission scenario, are developed that allocate the functions to the physical components. These operational flow diagrams represent system tasks that are stimulus/response oriented. The stimuli that can cause system tasks to be initiated are the operator initiated stimulus (operator depresses a key), the event initiated stimulus (receipt of messages on a data link or hardware interrupts), and the cyclic stimulus (computing navigation position at a 25 hertz rate). Once system tasks are identified, the flows that affect system performance, processor loading, operator workload, system reliability and maintainability are developed. The flows are then individually analyzed to see how a single task effects processor loading, data bus loading and system performance. These individual flows are then used in a

mission scenario to continue performing trade studies at a system level and to further refine the requirements. The types of analysis performed and required disciplines are detailed in Table 1.

Analysis	Disciplines
1. Allocation of performance to functional areas	Systems Engineers, Software Engineers, Hardware Engineers
2. Allocation of function to physical elements	Systems Engineers, Software Engineers, Hardware Engineers
3. Operator and System workload analysis	Systems Engineers, Software Engineers, Hardware Engineers, Human Factors
4. System Testability	Systems Engineers, Test Engineers
5. System Maintainability and Reliability	Systems Engineers, Maintainability, Integrated Logistic Support, Reliability

Table 1. (NU) Operational Analysis and Discipline Support

Figure 5 illustrates a mission scenario and two lower level system tasks that need to be performed during the engagement phase of the mission. Note that the two tasks have an overlap indicating that there is a need for parallel processing. As indicated in the figure, the functional processing has been allocated across the physical components. In addition, performance has been allocated to the functions to meet an overall performance requirement for the system. Computer based tools can now be used to analyze each task individually. Figure 6 illustrates an analysis of the Missile/Rocket Solution and shows that processor number 2 is 89.2% loaded with just this single task. This indicates that the allocation of function to the physical components, in this case, may not be optimal. Tools can assist in the reallocation of function by automatically analyzing alternate allocations of functions to the physical components. This analysis technique can now be extended to the entire system by modeling all tasks that are performed at any point in the mission scenario. Figure 7 illustrates the Missile/Rocket Solution task run in combination with the Gun Pointing Solution task. Note that there was a reallocation of function such that processor 2 is loaded at 54.6%.

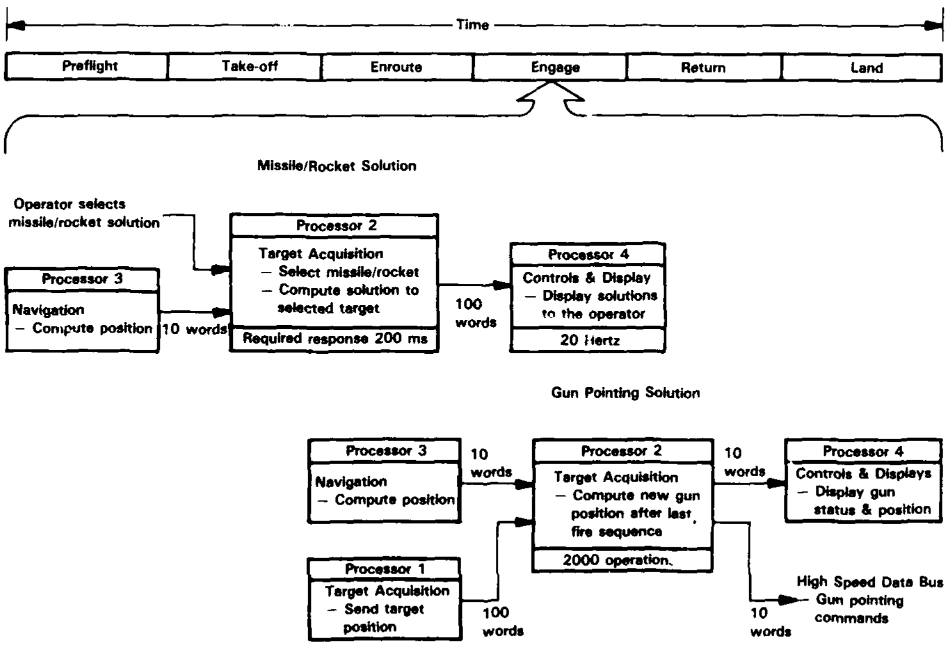


Figure 5. (NU) Operational Analysis of the Functional and Physical Approaches

Missile/Rocket Solution

Utilization Statistics of Hardware Components:

Resource	% Utilized	Resource	% Utilized
Processor 1	-	Processor Bus	4.0
Processor 2	89.2	HSDB	-
Processor 3	38.2	1553 Bus	3.8
Processor 4	6.9	Sensor Interface	32.9
Processor 5	-		
Processor 6	-		
Processor 7	-		
Processor 8	-		

Figure 6. (NU) Computer Analysis of the Missile/Rocket Solution Task

Combined Missile/Rocket Solution

Utilization Statistics of Hardware Components:

Resource	% Utilized	Resource	% Utilized
Processor 1	3.0	Processor Bus	12.0
Processor 2	54.6	HSDB	30.9
Processor 3	38.2	1553 Bus	33.8
Processor 4	67.8	Sensor Interface	32.9
Processor 5	60.8		
Processor 6	-		
Processor 7	-		
Processor 8	-		

Figure 7. (NU) Computer Analysis of the Missile/Rocket Solution Task in Combination with the Gun Pointing Solution Task.

Once the functional requirements and physical design are determined to be suitable using the operational model, the incremental level of the system development has been completed. An assessment of the overall system to date is made, and the iterative development process then begins again.

4.1 (NU) THE SYSTEM DEVELOPMENT PHASES

SET establishes control over the system development phases by:

- Use of a repetitive procedure that is applied to each incremental level of system definition
- Use of a plan to manage and coordinate each incremental level of system definition
- A tightly controlled exit from each incremental level of system development

To develop complex systems, it is advantageous to use a methodology which is essentially the same for each level of the incremental development of the system, since it is important that one incremental level flow smoothly into the next. This can only be effected through clearly defined, standardized models and procedures which pick up development where the last iteration leaves off.

Figure 1 shows major systems engineering specification activities throughout the system development phases of a program. Each increment of system definition has a unique set of activities and objectives culminating with a system level review milestone. SET applies the same core procedure iteratively at each increment of system definition. This procedure is based upon the use of the three models, and is specialized via a set of development criteria suited to meet the objectives of the individual increment.

SET employs a Systems Engineering Management Plan (SEMP) to control the overall system development process. The management plan provides for control over the technical development, the engineering specialty integration, and the iterative systems engineering development. Since SET features clearly defined end points for each incremental level, the iterative development of the system can be crisply controlled by a management plan. Just as the military standards provide for baseline control points via the system level reviews, SET provides control points via the inspections between the system review milestones. The inspection provides a controlled exit to ensure that all aspects of the system have been developed before the next incremental level of the system is defined.

It is utilized to review technical correctness, to review and update schedules and plans, and to baseline the system at the current level of development.

At the completion of each incremental development level, SET procedures include an evaluation step which involves the use of systems engineering measurements. These measurements are assessments of both the system being developed and the method being used. SET procedures specify an evaluation step for assessment in two measurement categories: quality and productivity. Quality measurements serve to evaluate the effectiveness of the system in meeting the requirements. Productivity measurements serve to evaluate the effective use of available resources in developing the system.

5.1 (NU) SYSTEMS ENGINEERING MEASUREMENTS

The systems engineering measurements are viewed by SET as an integral part of the development process. The formalization of systems engineering measurements is a relatively new effort and, to date, the measurement of quality has received the most emphasis. SET developers are currently focused on clearly defining systems engineering measurements for both quality and productivity aspects of development, along with their respective nominal values. This ability to evaluate the system development process provides SET developers with information which can be used to refine the methodology itself.

Measurements are taken and evaluated both within an incremental development level and throughout the system development phases. For each incremental level, the measurements are directed toward measurement of the system as it exists at that particular point in development. Throughout the system development phases, a synthesis of measurements is an ongoing process, necessary in evaluation of the system as a whole. When a measurement is identified as being less than optimal, the source of the problem is identified as the customer, the user, or the systems engineering organization. If the source is systems engineering, an improvement to the process is effected. If the source is the customer or the user, the measurements provide a vehicle for discussion of the problem.

One measurement currently used to track specification quality is the number of paragraphs changed in the specification against the major program milestones. Since the cost of fixing errors increases over time (see Figure 8), a desirable trend in the data should be a constantly decreasing number of errors found over the system development phases. Initial data indicates that SET is making a difference in specification quality. Figure 9 is a plot of paragraphs changed for a Software Requirements Specification (SRS) on two programs developed at IBM in Owego, New York. Although the avionics systems were different, they were developed by essentially the same team of engineers and were of the same order of magnitude (125,000, 16-bit words). The data shows that the specification developed with SET (Program B) is a marked improvement over the specification that was developed before SET was available (Program A).

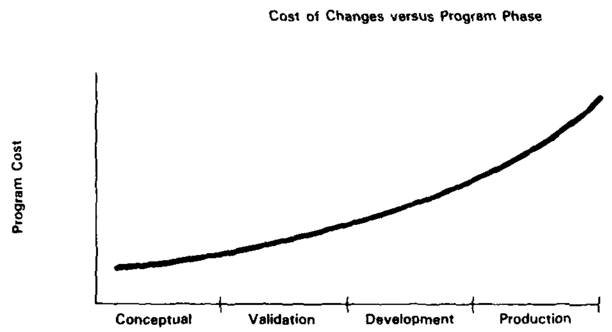
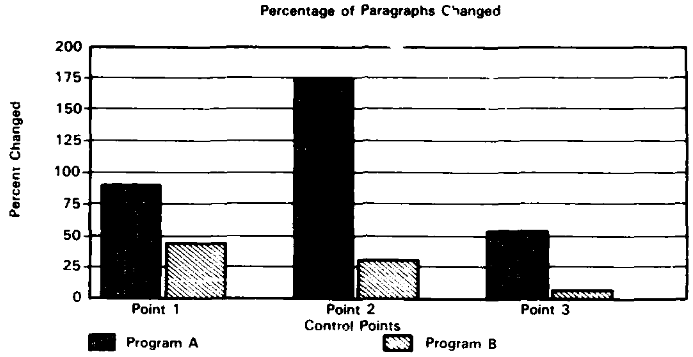


Figure 8. (NU) Cost of Fixing Errors Versus Program Phase



Control Points.

Point 1: Changes made between release of the specification by Systems Engineering and completion of design by Software Engineers

Point 2. Changes made between completion of software design and lab test

Point 3: Changes made between lab test and formal demonstration.

Figure 9. (NU) Systems Engineering Measurement Data

DISCUSSION

P.Simons, US

(1) How is the tool mechanized, and where can I learn more?

(2) What criteria are used in the early stages for the physical model?

Author's Reply

(1) PSL/PSA is used for recording and analyzing the functional requirements. A PC-based tool is used for the operational model analysis. This PC based tool is under development and is program unique at this point. For more information, contact Bruce Radloff, Manager, Systems Engineering Technology, IBM, Owego, NY (607) 751-2121.

(2) Early in system definition, criteria are derived by considering the mission scenario(s) and first-level functional requirements. If there is no existing hardware, new hardware is specified. Careful consideration must be given to the state of the technology, or predicted technology in this case.

P.Aouad, CA

(1) What is the typical time span of the development cycle?

(2) What is the time span to develop the:

- Functional model?
- Physical model?
- Operational model?

Author's Reply

(1) An average of 6 to 8 weeks is needed for each level of system definition. Less time is needed initially, and more time is needed at the lower levels.

(2) It is difficult to assign specific times to each model, since the process is not serial. The development team will typically alternate between the models until their impacts on each other are minimized or eliminated. The time to develop each model also depends on the system. For example, a system built using off-the-shelf hardware requires less time in the development of the physical model than one requiring new hardware.

J.Shepard, UK

You mentioned the use of PSL/PSA to store data from the models. You also said that the models, apart from the functional model, were manual, not automated. How are these data entered into PSL/PSA? Do you have an automated capture technique?

Author's Reply

The data on past programs were entered manually using templates resident on a computer or on paper. A database entry technician enters the information into the database. Functional requirements are then automatically produced from the database. We have a prototype tool that allows the engineer to construct an information flow diagram. From this diagram, the tool automatically produces PSL templates.

G.Bouche, GE

Is your procedure an engineering goal for future systems engineering tasks, or did you already complete a practical project using the procedure? If so, what type of project was it?

Author's Reply

The procedure has been used on Combat Talon II, V-22, and LHX/ARTI. We use this procedure on all of the avionics programs, and it is becoming our division standard. We also have a three-day workshop on its use.

W.H.McKinlay, UK

(1) With what level of detail and flexibility are scenarios defined?

(2) How do you verify that the correct match between man and sensors/systems is achieved?

Author's Reply

(1) Initial mission scenarios are defined to the extent necessary to understand the system characteristics and its environment. Many scenarios may l necessary. All flows in the operational model are derived from the initial mission scenarios and:

- Are operator-initiated.
- Have critical performance requirements (e.g., timing and accuracy).
- Affect system loading, operator loading, and system moding.

(2) The operator-initiated flows and flows that affect the operator workload allow us to establish the relationship between the operator and the system.

MAQUETTAGE DES SPECIFICATIONS FONCTIONNELLES

DU LOGICIEL EMBARQUE

EXPERIENCE DU SYSTEME AVIONIQUE RAFALE

Monsieur Patrick SCHIRLE
Ingénieur d'études à la direction générale technique

AVIONS MARCEL DASSAULT
78, Quai Marcel Dassault
92 SAINT-CLOUD - FRANCE

RESUME

Le développement du logiciel des systèmes avioniques requiert une documentation de spécification fonctionnelle abondante et souvent contractuelle. Le maquettage de ces spécifications permet :

- d'améliorer la qualité formelle des spécifications (cohérence, complétude, lisibilité)
- de réaliser, très tôt dans le cycle de vie, une validation fonctionnelle des spécifications sur simulateur (miroir de la spécification)
- de fournir des éléments de recette (jeux d'essais) aux réalisateurs de sous-ensembles du système
- de disposer d'une référence fonctionnelle commode lors de l'intégration des matériels réels.

Le maquettage intervient dès la phase de définition fonctionnelle du logiciel selon le scénario suivant :

- première écriture des spécifications fonctionnelles du logiciel
- analyse critique des spécifications : contrôle de forme
- codage des spécifications dans l'ordinateur de simulation de la maquette
- tests fonctionnels sur maquette : contrôle de fond
- fourniture aux réalisateurs des équipements de spécifications réputées bonnes et des jeux d'essais correspondants.
- réalisation du logiciel des équipements
- intégration et validation du système réel

Le système avionique de l'avion RAFALE présente des innovations telles que l'intégration des systèmes avion (moteur, commandes de vol, etc...) ou la sécurisation des informations de pilotage. Ces nouvelles fonctions entraînent un accroissement et une évolution qualitative notables du logiciel des calculateurs embarqués.

Pour améliorer la qualité et le délai de mise au point des logiciels, le développement s'est appuyé sur une méthodologie intégrant le maquettage des spécifications fonctionnelles du logiciel.

La maquette a été construite autour d'un calculateur SEL supportant le codage Fortran des 3500 pages de spécifications fonctionnelles du système.

Chaque spécificateur a pu valider sa spécification, puis globalement l'ensemble des spécifications, directement sur maquette. A cette fin, un conversationnel spécifique a dû être développé pour permettre la stimulation du logiciel maquette et l'exploitation des résultats grâce à des scénarios représentatifs des conditions d'utilisation du système réel.

Outre le gain réalisé pour l'intégration du système réel, la maquette a pu être utilisée comme :

- support d'analyse de pannes
- banc d'essais pour les évolutions
- générateur d'éléments de recette
- référence fonctionnelle pour les intervenants

1 - INTRODUCTION

Les systèmes avioniques actuellement développés par la Société des Avions Marcel Dassault se caractérisent par :

- Une complexité croissante
- Une intégration de plus en plus serrée de leurs fonctions
- Une inflation chronique du volume de logiciel embarqué.

Ceci implique la coordination des travaux de très nombreux spécialistes et une organisation industrielle adéquate : face à ces problèmes et pour assurer la qualité constante du produit, la solution actuellement retenue par l'avionneur consiste en la définition, la mise en place et le suivi d'une méthodologie stricte concernant le développement du système, particulièrement du logiciel embarqué.

Cette méthodologie précise sans ambiguïté quelles sont les étapes du développement, les produits et les responsabilités qui leur sont attachés. Liée au traditionnel cycle de vie du logiciel, elle fait apparaître des phases de conception/spécification, de réalisation (codage) et d'intégration/validation.

La phase de conception/spécification se révèle comme étant la plus critique. En effet, s'inscrivant dans la partie amont du cycle de vie, toute erreur ou imperfection à ce niveau est amplifiée au cours du cycle et se traduit par des conséquences avales coûteuses et difficilement maîtrisables.

De plus, les produits issus de cette phase étant essentiellement des documents papiers, la perception du produit final à travers ces simples documents est délicate et problématique.

Les deux étapes maîtresses de la phase de conception/spécification sont :

- L'étape de spécification globale du système
- L'étape de spécification fonctionnelle détaillée

L'étape de spécification globale consiste en l'écriture de deux types de documents :

- Règles générales : il s'agit d'une description des règles et philosophies d'emploi du système, communes à toutes les missions (principe de dialogue homme-machine, gestion/signalisation des pannes, superposition des fonctions d'armes, etc...).
- Spécifications globales des fonctions opérationnelles : il s'agit de décrire, pour chaque fonction spécifique du système, le scénario nominal d'utilisation de cette fonction, en terme d'utilisateur (pilote) et bien entendu dans le respect des règles générales.

Ce type de spécification, non exhaustif sur un plan fonctionnel et ne dédiant pas les rôles respectifs de chacun des équipements, permet de valider en terme d'opérationnel la conception des fonctions et sert de base à l'étape suivante.

La validation de cet ensemble de spécification est réalisée pratiquement grâce à un simulateur simplifié construit autour d'une cabine de pilotage, permettant un dialogue direct et aisé avec les pilotes.

L'étape de spécification fonctionnelle détaillée se compose de deux tâches :

- Etablissement d'une architecture fonctionnelle : décomposition a priori en modules des tâches à réaliser et répartition de ces modules dans les équipements ; principes généraux de dialogue entre équipements et entre modules d'un même équipement.
- Ecriture des spécifications fonctionnelles détaillées par module et par équipement, description exhaustive des traitements à effectuer et des interfaces.

Ces documents représentent la dernière étape de la phase de conception directement du rôle et de la responsabilité de l'avionneur : ils serviront de référence contractuelle vis à vis des coopérants pour la réalisation du logiciel des équipements.

C'est à ce niveau de spécification que nous ressentons un besoin constant d'amélioration de la qualité, tant formelle que fonctionnelle.

Les méthodes classiques que nous utilisons depuis plusieurs années pour "valider" ces documents (représentant plusieurs milliers de pages) sont :

- Canevas strict
- Organisation de relectures croisées
- Dictionnaire de terminologie
- Traitement de texte (texte + graphique)

Nous tentons actuellement une ouverture vers des méthodes exploratoires plus "up-to-date" qui sont :

- L'utilisation d'un langage de spécification formel et contrôlé par outil.
- Le maquettage des spécifications fonctionnelles détaillées.

Le langage formel n'est employé à ce jour que pour des applications spécifiques (Commandes de vol, chaînes à haute sécurité) et n'a pu être systématique ; en effet les langages actuels sont ciblés sur un type particulier de spécification et requièrent une formation spécifique. Etant actuellement trop éloignés du langage naturel pour permettre une lecture (ou une écriture) par un nombre important d'individus de fonctions et de compétences diverses, leur utilisation quoique inévitable à terme, fait actuellement l'objet d'études.

Le maquettage fonctionnel par contre présente, outre sa faisabilité démontrée par les méthodes et tactiques actuelles, l'avantage d'assurer à la fois la qualité formelle de la spécification (par nécessité) et la qualité fonctionnelle de la même spécification par effet de miroir.

Cette stratégie de maquettage a été retenue et mise en pratique dans le cadre du programme RAFALE à l'étape d'écriture des spécifications fonctionnelles détaillées.

2 - CONTEXTE RAFALE : HYPOTHESES ET SPECIFICITES

2.1 - Evolution des spécifications fonctionnelles détaillées

Pour les systèmes des premières années 80, il était réalisé un unique document de spécification par équipement, lequel décrivait globalement l'ensemble des fonctions de chaque équipement sans contrainte particulière de modularité. Ce document débouchait sur la réalisation du logiciel équipement correspondant et toute modification du premier entraînait ipso-facto modification du second selon une gravité variable et non prévue.

Pour des raisons de taille et d'évolutivité du logiciel et de la documentation, il est devenu nécessaire à partir des systèmes MIRAGE 2000 EXPORT d'affiner la résolution du couple document de spécification/logiciel grâce à la création d'une architecture fonctionnelle du logiciel des équipements.

Cette architecture représente la découpe en un certain nombre de modules indépendants, des fonctions à réaliser par le logiciel d'un équipement. Cette découpe, permettant le cloisonnement entre des modules bien identifiés, s'appuie sur des critères :

- d'évolutivité :
  modules à faible/forte probabilité d'évolution.
  Exemples : forte probabilité : conversationnel homme-machine
  faible probabilité : algorithmes de balistique

- de sécurité :
  modules de différents niveaux de criticité (cf. DO 178)

- de récupérabilité :
  modules spécifiés comme récupérables d'un système à l'autre (module indépendant de l'environnement spécifique d'un système donné).

L'architecture fonctionnelle, établie par AMD-BA pour l'ensemble des équipements est prise en compte pour l'écriture des spécifications et débouche sur une modularité fonctionnelle de la documentation et in fine du logiciel correspondant : à chaque document de spécification correspond un module identifié du logiciel :

Caractéristiques des spécifications "modulaires" :

- Chaque module est spécifié par un et un seul document de spécification.
- Chaque module est défini, géré, modifié de façon autonome.
- Chaque module est décrit par :
  . ses interfaces avec les autres modules
  . sa fonction de transfert
- Chaque module peut être réalisé de façon autonome
- Chaque module est décomposé en sous-modules et à terme en pièces de logiciel.

2.2 - Contexte RAFALE

Le système avionique de l'avion démonstrateur RAFALE présente un certain nombre de nouveautés et de spécificités par rapport aux avions de la génération précédente, telles que :

- l'intégration très poussée incluant les systèmes avion
- l'extension du nombre de fonctions nécessaires dès le premier vol
- la décentralisation des fonctions système (réparties dans plusieurs équipements)
- la généralisation de l'emploi des techniques numériques dans des domaines où l'expérience en était faible
- les délais très courts et très tendus de l'opération.

Pour faire face à cette situation, il a alors été décidé de réaliser un travail de maquettage des spécifications avec les objectifs suivants :

a) Améliorer la qualité des spécifications fonctionnelles détaillées

b) Permettre une validation fonctionnelle de ces spécifications

c) Fournir aux coopérants des jeux d'essais cohérents pour valider aussi tôt que possible leurs développements.

d) Disposer d'un banc d'essai pour tester a priori les évolutions.

Ce travail a débuté en Mars 85 avec les étapes suivantes :

a) Analyse critique des spécifications

b) Réalisation de la maquette

c) Exploitation de la maquette

3 - ANALYSE CRITIQUE DES SPECIFICATIONS

3.1 - But

Les spécifications fonctionnelles détaillées représentent la charnière entre la conception (travail AMD-BA) et la réalisation des logiciels (travail des coopérants). Il est donc important qu'elles soient à la fois :

- Entièrement représentatives des besoins opérationnels du concepteur (aspect fonctionnel)
- Compréhensibles et réalisables par les coopérants (aspect formel)

Le but de l'analyse critique des spécifications est d'améliorer leur qualité pour couvrir l'aspect formel, c'est-à-dire s'assurer que les spécifications sont :

- lisibles
- complètes
- cohérentes entre elles
- sans ambiguïté
- réalisables informatiquement

3.2 - Principe et organisation

Un spécificateur étant naturellement satisfait de son document grâce à sa connaissance du contexte opérationnel, pour que l'expérience soit rentable il a fallu isoler les lecteurs critiques de ce même contexte en limitant les explications fournies sur les spécifications. Le mot d'ordre a été : ne pas juger de ce que doit faire la spécification (fonctionnel) mais juger uniquement la manière dont elle est écrite et sa faisabilité.

L'équipe de relecture n'a pas eu connaissance du besoin opérationnel exprimé à travers les spécifications détaillées (spécifications globales non fournies).

la totalité des documents de spécification détaillée (3000 pages) a été soumise à l'équipe de relecture critique. Toute fiche d'évolution, quelle que soit son origine, s'est vue appliquer la même procédure de relecture.

3.3 - Résultats de l'étape

Le nombre de critiques (justifiées) a été extrêmement important : 250 pages de remarques, représentant de l'ordre de 1000 points précis.

Le nombre de critiques par page de spécification (donc in fine la qualité formelle de la spécification) varie considérablement en fonction :

- du rédacteur (rigoureux/non rigoureux, précis/général, structuré/collectionneur de détails)
- du module spécifié (logique/algorithmique)

Les erreurs relevées par la critique entrent toutes dans les trois catégories :

- erreurs de rigueur
- erreurs de généralité
- inadaptation de la spécification à une réalisation informatique.

a) Erreurs de rigueur

- Interface manquante : l'information utilisée par les traitements n'est pas déclarée à entrée de la spécification
- Interfaces incohérentes : les interfaces déclarées en entrée du module spécifié n'existent pas dans le système (non calculées par d'autres modules)
- incomplétude des traitements
  . Le traitement relatif à une sortie déclarée du module n'est pas spécifié
  . Dans une combinaison logique, tous les cas ne sont pas renseignés ; il est à noter que le cas est très fréquent lorsque la logique est exprimée au moyen de phrases (si, alors, sauf, quand) et pratiquement inexistant si la logique est décrite sous forme de tableaux de vérité.
  . Les conditions d'initialisation, d'activation, d'enchaînement des traitements ne sont pas spécifiées.
- Terminologie floue ou ambigüe
  Exemple :
  "on déterminera ..."
  "dans la plupart des cas..."
  "dans certaines conditions..."
  "l'information existe dans les cas suivants..."

b) Erreurs de généralité

- Caractéristiques d'interfaces non spécifiées

  Ne sont pas précisés l'unité, le type (logique, booléen), les valeurs possibles d'une information.

- Traitement décrit trop globalement :

  Ce genre d'erreur est fréquent lorsque le spécificateur surestime le savoir faire (ou l'intuition) des réalisateurs de logiciel.

c) Inadaptation de la spécification à une réalisation informatique

- Choix de la solution informatique : le spécificateur, dans un louable souci de rigueur impose la façon dont doit être réalisé le traitement : connaissant peu les critères de "programmation", le choix est parfois non astucieux et peut déboucher sur un logiciel démesuré ou non évolutif.

3.4 - Remarques et commentaires sur l'étape d'analyse critique

L'étape d'analyse critique des spécifications a été extrêmement fructueuse et révélatrice. Nombre de problèmes de toute importance ont pu être ainsi résolus a priori, évitant de les reporter à la phase d'intégration. Par contre, l'énergie mise en oeuvre a été également importante : équipe de relecture, temps "volé" aux spécificateurs, lourdeur de mise à jour de la documentation.

La plupart des erreurs recensées sont des erreurs évitables qui ne remettent pas en cause le profil actuel des spécificateurs. En effet, cette phase de relecture a conduit à améliorer une spécification existante, non pas à créer une couche de spécification plus détaillée.

Enfin, l'expérience a été vécue par les spécificateurs comme un contrôle qualité supplémentaire imposé, donc ressentie de façon très mitigée...

4 - REALISATION DE LA MAQUETTE

4.1 - Elaboration des programmes-maquette

La nécessité de coder les spécifications a mis en évidence trois impératifs d'écriture de celles-ci. Les deux premiers sont d'ordre général et concernent toute spécification devant être codée, le troisième est lié à la structure doublée du système RAFALE.

Premier impératif : complétude des interfaces
Deuxième impératif : définition du type de chaque variable
Troisième impératif : définition pour chaque variable du type de liaison entre module émetteur et module(s) récepteur(s)

D'autre part, l'opération de codage a confirmé la nécessité de description d'un logiciel enveloppe pour chaque module implanté dans un équipement doublé.

Complétude des interfaces

Cette étape est indispensable avant toute opération de codage. Le renseignement complet des interfaces a donc été nécessaire, avant codage du logiciel initial, comme avant chaque passage d'une version à l'autre.

Définition du type de chaque variable

Cet impératif a conduit à enrichir la base de donnée d'interfaces avec la définition, pour chaque variable, d'un type analogue aux déclarations de variables FORTRAN, à savoir :

- booléen, tableau de booléens, logique, réel, entier.

Nota : La phase d'analyse critique des spécifications avait déjà fait apparaître la nécessité de définition du type de variable.

Codage des modules fonctionnels

Les programmes maquette ont été écrits en FORTRAN, directement à partir de la spécification (après phase d'analyse critique) et en utilisant la base de donnée d'interfaces comme référenciel des variables.

Produits

Le résultat de la phase d'élaboration des programmes-maquette se décompose en :

- Un produit intermédiaire sous la forme d'un fichier de variables de 8 caractères extrait de la base de données d'interfaces. Ce produit constitue en fait un complément de spécification, indispensable pour la génération du code.

- Un produit final composé des programmes-maquette (ou modules) d'une chaîne de calcul. L'obtention de l'ensemble des modules représentatifs des deux chaînes devant être faite grâce à la duplication de ces programmes-maquette.

4.2 - Adaptation de la console de visualisation des échanges (CVE)

La CVE était, à l'origine, un outil de visualisation des échanges entre équipements numériques. Pour les besoins du maquettage, il a fallu faire tendre cet outil de visualisation vers un outil de validation de spécifications. Les travaux d'adaptation ont porté sur :

Au niveau conversationnel

- Le pilotage de la simulation (choix du mode simulation, des modules à activer, etc...)
- Le développement de procédures d'entrée de valeurs à la CVE
- L'amélioration de la gestion des chaînes mémorisées (concaténation de chaînes, appel nominatif de celles-ci).
- Le développement des procédures de tests automatiques

Au niveau système

- La génération des lexiques CVE

Ces lexiques sont nécessaires au bon fonctionnement de la simulation et à l'exploitation des résultats de celles-ci.

Ils comprennent :

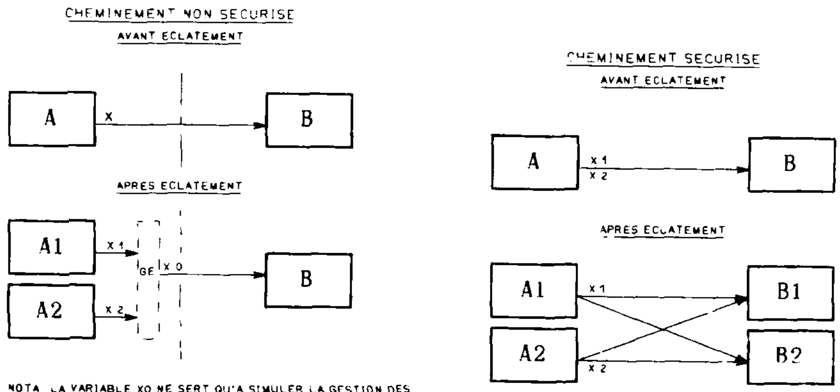
. la liste des modules et leurs adresses
. la correspondance des fichiers 8 caractères par rapport aux fichiers de la base de donnée d'interface initiale (40 caractères)
. la liste des pavés récepteurs d'une information donnée

4.3 - Mise en oeuvre de la simulation

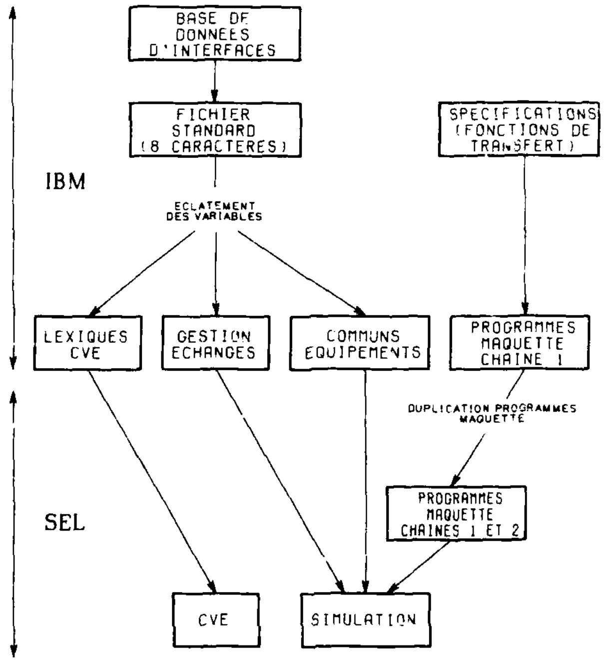
Les caractéristiques essentielles de la simulation mise en place sont les suivantes :

- Simulation mono-fréquence
- Cycle de simulation correspondant à l'activation séquentielle des modules maquettés
- Prise en compte de l'architecture double-chaîne du RAFALE :
  Cette prise en compte se résume à l'établissement de deux procédures :
  . Eclatement des variables
  . Duplication des programmes

Afin de générer l'ensemble des variables émises et reçues dans les équipements des chaînes 1 et 2, il a été nécessaire, à partir du fichier standard 8 caractères, d'éclater les variables pouvant être émises par deux équipements symétriques (voir schémas ci-dessous).



NOTA LA VARIABLE XO NE SERT QU'A SIMULER LA GESTION DES ECHANGES DANS SON ROLE D'AIGUILLAGE DES VARIABLES EMISES ALTERNA[illegible] A1 ET A2



RECAPITULATIF DES PHASES DE LA REALISATION MAQUETTE

5 - 1ère EXPLOITATION MAQUETTE - TESTS UNITAIRES ET MANUELS

Définitions :

Tests unitaires : tests portant sur les variables d'Entrées/Sorties d'un module unique
Tests manuels : tests pour lesquels les valeurs des variables d'Entrées doivent être modifiées manuellement par le spécificateur.

Le déroulement de ce type de test est le suivant :

1) Constitution d'une chaîne
2) Entrée des valeurs à la CVE
3) Déclenchement d'un pas de simulation et lecture des résultats

5.1 - Constitution d'une chaîne

Cette opération consiste à sélectionner un certain nombre de variables parmi le total des variables d'E/S d'un module donné et à les regrouper dans un ensemble appelé chaîne. Suivant la taille de ce module ce choix a été fait :

Pour les modules de taille importante

En sélectionnant toutes les variables d'Entrées/Sorties se rapportant à une entité donnée.

Pour les modules de taille réduite

La taille des sous-modules de ces spécifications permet d'utiliser toutes les variables de ces sous-modules pour constituer les chaînes. Chaque chaîne est l'image d'un sous-module.

Cette méthode permet :

- de retrouver chaque variable de sortie du module dans une chaîne
- de disposer dans chaque chaîne de toutes les informations utilisées pour élaborer les variables émises par le sous-module.

On peut ainsi se rapprocher de la démarche visant à valider les pièces de spécification de niveau le plus bas avant de passer au niveau supérieur.

5.2 - Entrée des valeurs à la CVE

Cette opération est réalisée en désignant la variable à modifier dans la liste des variables de la chaîne, en sélectionnant le mode "modification de valeur", puis en entrant la nouvelle valeur.

Bien que la modification manuelle de valeur de n'importe quel type de variable soit possible, la plupart des tests ont porté sur des modifications de variables booléenns VRAI/FAUX, et de variables logiques.

Quelques tests ont porté sur des modifications de variables numériques soit pour vérifier des logiques (déclenchement de seuils, temporisations...), soit plus rarement, pour valider des fonctions de transfert numériques.

6 - 2ème EXPLOITATION MAQUETTE - TESTS AUTOMATIQUES

6.1 - Intérêt des tests automatiques

L'utilisation des tests manuels a révélé plusieurs limitations de ces derniers :

1) Difficultés de manipulation des chaînes comportant un nombre important de variables.
2) Inadaptation de ces tests pour la recherche de dépendances entre variables.
3) Difficultés de validation des mécanismes mettant en jeu des transitions ou des mémorisations.

Ces limitations ont amené à envisager le développement de tests automatiques qui permettraient à la fois :

- de générer automatiquement des combinaisons de valeurs d'entrée pour les chaînes à tester
- de faciliter l'exploitation des résultats par des éditions appropriées sur listings.

Génération automatique de valeurs d'entrée

Les objectifs visés correspondent aux limitations mentionnées plus haut pour l'exploitation des tests manuels :

- Pouvoir balayer toutes les combinaisons des variables d'entrée choisies et observer leur impact sur toutes les variables émises par le module pour détecter d'éventuelles dependances anormales entre variables.
- Mettre au point des scénarios nominaux permettant de simuler différentes configurations d'initialisation, des enchaînements de phases de vol mettant en jeu des transitions ou des mémorisations.

Les premiers tests ainsi mis en place appelés tests mono-variables ont permis de tester toutes les combinaisons déduites d'une combinaison initiale en faisant varier une variable booléenne de la chaîne.

La validation des transitions et mémorisations a donné lieu à la création du type de test "SCENARIO DE PANNE". Ce type de test permet d'introduire à des intervalles de temps traduits en nombre de pas de simulation des jeux de valeurs d'entrée préparés à l'avance.

La vérification exhaustive de tous les cas possibles de valeurs d'entrée d'une chaîne de variables booléennes donnée a été rendue possible grâce aux tests combinatoires.

En tout 7 types de tests automatiques ont été développés :

a) Combinatoire statique
b) Combinatoire séquentiel
c) Mono-variable statique
d) Mono-variable séquentiel
e) Aléatoire statique
f) Aléatoire séquentiel
g) Scénario de panne

Nota : Un test statique se distingue d'un test séquentiel par la remise de toutes les variables à la valeur d'initialisation après chaque pas de simulation.

Facilités d'exploitation des résultats

Parallèlement au développement de ces types de tests, les possibilités de sortie des résultats ont été étendues :

- Possibilités d'impression ou non de la "référence" (c'est-à-dire de la chaîne avec ses valeurs d'initialisation).
- Possibilité d'éditions sélectives des variables ayant varié par rapport au cycle précédent.
- Possibilité d'éditions sélectives des variables ayant varié par rapport à la combinaison de référence.

Ces possibilités ont permis en particulier de manipuler des chaînes constituées d'un grand nombre de variables, sans pour autant avoir à rechercher les résultats significatifs dans la liste de toutes les variables constituant la chaîne.

6.2 - Exploitation des tests automatiques

La phase d'exploitation des tests automatiques a démarré en automne 85 et a comporté deux parties :

- Tests unitaires automatiques
- Tests globaux automatiques

Tests unitaires automatiques

Suivant la nature de la spécification, deux types de tests ont été principalement utilisés :

- Tests combinatoires
- Scénarios de panne

Pour les spécifications décrivant des mécanismes de logique sans mémorisation ni problèmes d'initialisation, la majorité des tests a été du type combinatoire statique (spécifications de visualisations notamment).

Pour les spécifications décrivant un nombre important d'états, de transitions et de temporisations, des scénarios de panne ont été généralement employés (spécification de signalisation des informations moteur ou commandes de vol notamment).

Tests globaux

Définition
- Test global : test consistant à activer l'ensemble des modules maquettés.

Nota : Ce type de test, disponible également en mode manuel n'a été utilisé pratiquement qu'en mode automatique.

Intérêt

L'intérêt des tests globaux est de valider le comportement de l'ensemble des modules maquettés.
En effet chaque spécification peut être vérifiée en théorie, à la seule lecture du document de spécification lui-même.

Par contre, il n'existe pas de document décrivant la répartition précise des traitements entre les différentes spécifications et assurant ainsi que la mise bout à bout des différents modules conduise au respect de la spécification globale.

7 - BILAN

7.1 - Ressources spécifiques mises en oeuvre

L'analyse des spécifications et leur codage dans la maquette ont requis 50 Homme-Mois, dont près de 50 % pour la phase d'analyse critique des spécifications. Le développement de la chaîne d'outils ainsi que la mise en oeuvre de la maquette ont nécessité 22 Hommes-Mois.

Sur le plan matériel, la maquette a été réalisée sur un système informatique GOULD (SEL 32/27) dont il a été nécessaire d'augmenter la puissance au cours du développement en raison de la dégradation du temps de réponse. Il faut souligner que la réalisation de la maquette a pu se faire dans les temps impartis grâce à l'utilisation de l'expérience acquise lors de développements antérieurs dans le domaine de la simulation.

7.2 - Répercussions sur la méthode de travail

L'introduction du maquettage a modifié la méthode suivie pour développer le logiciel avionique RAFALE par rapport aux habitudes des programmes précédents. Cette modification concerne essentiellement :

- la déclaration des interfaces entre modules (y compris modules d'un même équipement)
- l'écriture d'une spécification avec une contrainte de maquettabilité
- la possibilité de voir "vivre" une spécification

Déclaration des interfaces entre modules

La cohérence imposée au niveau des interfaces par la saisie de celles-ci sur l'outil de base de données (première étape du maquettage) a induit une plus grande rigueur dans le dialogue entre spécificateurs. En effet l'obligation de rentrer chaque interface dans une référence unique et regroupant l'ensemble des interfaces a évité la plupart des redondances d'informations (ou les origines d'informations inconnues) que la dispersion des interfaces aurait risqué d'entraîner.

Ecriture des spécifications avec une contrainte de maquettabilité :

La nécessité de décrire des traitements pouvant être transcrits sans ambiguïté en code exécutable (en l'occurrence FORTRAN) a permis d'éviter des retards dus aux difficultés rencontrées par les coopérants dans la lecture de spécifications détaillées trop "générales".

Possibilité de voir vivre une spécification

Lors d'application de fiches de modifications, la possibilité de valider presque immédiatement celles-ci a permis de "resserrer" le lien entre le spécificateur et son produit. En effet, bien qu'il soit pratiquement toujours possible de tester une modification (ou un mécanisme de façon générale) mentalement ou sur le papier, l'utilisation d'un outil accomplissant lui-même l'effort nécessaire à l'exécution de la fonction de transfert a déchargé d'autant le spécificateur, lui permettant ainsi de se consacrer à l'interprétation des résultats.

Par contre le bilan de l'opération maquettage fait apparaître des contraintes ayant limité l'intérêt de celle-ci, en ce qui concerne le programme RAFALE. Ces contraintes sont de deux types :

- Ergonomie de la maquette
- Délais et disponibilité des utilisateurs

Ergonomie de la maquette

Malgré les efforts importants d'aménagement du conversationnel de la maquette, la présentation très désincarnée des informations a rapidement modéré l'empressement des utilisateurs pour ce nouvel outil. Il est en effet difficile de valider le comportement d'un ensemble de réticules (à plus forte raison d'une page complète de réticules), à l'aide de VRAI/FAUX ou de variables logiques codées. Cette présentation quelque peu rébarbative n'a pas permis d'exploiter complètement les possibilités pourtant importantes de la maquette.

Délais et disponibilités des utilisateurs

Les périodes de disponibilité des utilisateurs de la maquette n'ont pas toujours coïncidé avec les périodes où il aurait été le plus profitable d'exploiter celle-ci. En particulier les tests globaux, intérêt majeur de la maquette, n'ont pas eu l'importance qu'ils auraient dû avoir, en raison de l'exploitation tardive de ceux-ci.

7.3 - Répercussion sur la qualité du logiciel avionique

L'analyse de l'impact du maquettage sur la qualité du logiciel fourni aux bancs d'intégration est rendue difficile pour les deux raisons suivantes :

- Le maquettage est l'une des composantes de la méthode de travail, au niveau des spécifications fonctionnelles détaillées ; le résultat obtenu est donc inhérent à l'ensemble de la méthode, l'impact purement maquette étant quasiment impossible à extraire.
- La nouveauté des fonctions assurées par le logiciel, la nouveauté de l'architecture et le caractère d'avion démonstrateur font du Système Avionique RAFALE un cas particulier, rendant toute comparaison délicate par rapport aux systèmes précédents.

Néanmoins, analyse faite par le personnel des bancs d'intégration, il ressort que le nombre de fiches d'évolution liées aux erreurs du niveau spécification est en nette régression par rapport à son niveau habituel ; relativement à l'ensemble des fiches d'évolution, sa part a été réduite environ de moitié.

8 - CONCLUSIONS ET ENSEIGNEMENTS DU MAQUETTAGE DE SPECIFICATIONS FONCTIONNELLES DETAILLEES

8.1 - Ce que le maquettage permet

a) Amélioration des spécifications

Contrôle de forme

La relecture critique permet de corriger et de compléter la spécification pour l'amener à un niveau de qualité autorisant son codage direct par des informaticiens (équivalent manuel des futures spécifications formelles où l'outil se charge de la relecture et permet une compilation).

Le maquettage agit à ce niveau comme un puissant révélateur d'erreurs et un contrôle qualité efficace.

Contrôle de fond

L'outil maquette permet une réelle validation fonctionnelle de la spécification au niveau module, équipement ou système : validation horizontale (tests exhaustifs d'un module) ou verticale (test d'une chaîne fonctionnelle donnée).

b) Fourniture de jeux de test ou d'éléments de recette

Les tests réalisés sur la maquette (combinaisons d'entrées et sorties correspondantes) ont servi à fournir des jeux d'essais aux coopérants.

Ces jeux d'essais ont été utilisés expérimentalement dans le cadre du RAFALE comme aide à la mise au point chez les coopérants.

En effet, la maquette étant représentative de l'architecture du système réel, elle permet de générer les informations de simulation ou de stimulation au niveau souhaité.

D'où la possibilité d'organiser, en amont de l'étape d'intégration une recette fonctionnelle partielle des équipements.

c) Disposition d'une référence fonctionnelle commode

Une fois validée, la maquette est entretenue de toutes les modifications apportées au système et reste donc la référence. Il est alors possible de l'utiliser (sous réserve d'une ergonomie suffisante) comme une "documentation vivante" et représentative du système, portable et ne nécessitant pas de matériel spécifique ; ce à des fins pédagogiques, didactiques ou trivialement pour analyse et contrôle de cas de fonctionnement non nominaux du système.

d) Visibilité au niveau évolution/récupération

La qualité des spécifications autorise l'analyse et le choix des évolutions, directement en terme de solution, permettant ainsi une connaissance a priori de leur impact sur le logiciel. La maquette elle-même peut servir de banc d'essai pour tester fonctionnellement les dites évolutions et s'assurer de leur adéquation aux besoins avant leur mise en chantier. Enfin, la correspondance document de spécification/module de logiciel permet de prévoir et de planifier la récupération de logiciel. Elle permet en outre, après analyse, de disposer de gabarits en matière de volume de logiciel, de charge calcul,... ou même coût.

e) Efficacité de la spécification

La conséquence directe du maquettage est de disposer de spécifications autonomes et auto-suffisantes pour réaliser et tester le logiciel, module par module ou équipement par équipement.

Pour des systèmes très décentralisés de type RAFALE, ceci permet de fournir à chacun des intervenants les éléments précis, nécessaires et suffisants à la tâche qu'il doit conduire. Ce point, qui concourt à la simplicité et à la clarté des documents, permet également une bonne protection des informations confidentielles.

f) Amélioration de la qualité du produit final

Outre le gain de qualité obtenu pour l'étape de spécification fonctionnelle détaillée, l'exploitation de la maquette, particulièrement l'utilisation des tests automatiques, permet et a permis d'améliorer la confiance dans le produit final. En effet des centaines de tests ont été réalisés relativement à des configurations avion non nominales, permettant ainsi de découvrir (et de remédier) à des situations non prévues a priori dans la spécification. De même la recherche d'un profil de pannes donné, par balayage combinatoire systématique de toutes les entrées a permis de conforter les analyses de panne et les solutions choisies.

8.2 - Ce que le maquettage implique

a) Une gestion rigoureuse et outillée des interfaces (18000 dans le cas du RAFALE)

La collection des interfaces entre modules et entre équipements représente le coeur de la "machine" ; c'est à partir de ces données que vont être créées automatiquement les variables logicielles de la maquette, lesquelles pourront être stimulées et scrutées en phase de validation.

Il faut donc disposer, avant de commencer le maquettage, d'une description complète et cohérente de ces interfaces, ce qui demande, outre une extrême rigueur, un gros travail de synthèse et de rapprochement.

b) Une description précise et complète des fonctions de transfert

Pour les modules que l'on désire valider, la réalisation du logiciel maquette correspondant consiste à décrire par du code exécutable (manuellement dans le cadre du RAFALE) les fonctions de transfert extraites du document de spécification. Il est donc indispensable que cette description soit suffisamment fine pour pouvoir exprimer chaque sortie du module selon une combinaison mathématique des entrées ; dans le cas où la description est trop générale, il y a valeur ajoutée entre la spécification et la maquette, ce qui est contraire au principe de validation des spécifications.

c) Une gestion de la documentation et de la configuration

L'abondance et la variété de la documentation, la nécessité de sa remise à jour systématique (toute évolution ne peut être décrite que par modification cohérente et immédiate de la documentation concernée), exigent un support de documentation souple et efficace.

De même, le parallélisme indispensable entre les états de définition :

- documentation (à tous niveaux)
- logiciel maquette
- logiciel réel

implique une gestion de configuration rigoureuse et commode.

8.3 - Conclusion

Pour synthétiser le bilan de l'expérience ainsi acquise, le mieux est de rappeler que l'avion RAFALE a effectué son premier vol le 4 Juillet 1986, avec six mois d'avance sur l'objectif fixé par l'Etat, et que les 90 vols effectués en six mois et l'évaluation conduite par le Centre d'Essais en Vol, l'Armée de l'Air et la Marine Nationale Française ont conclu à l'excellente adaptation de l'avion et de son système aux objectifs opérationnels qui lui avaient été fixés.

Les principes exposés dans cette note seront donc poursuivis en vue d'une application dans les programmes futurs, et d'une intégration dans l'effort méthodologique et de développement d'outils mené par les Avions Marcel Dassault et de façon plus générale par l'industrie aéronautique française.

DISCUSSION

E.Cambise, IT

(1) What is the size of the embedded software of the RAFALE?

(2) What is the ratio between the development effort of the prototype software and that of the "target" embedded software?

Author's Reply

(1) The size of the operational software, the specifications of which were submitted for prototyping, is of the order of 200 koctets. (This software does not include flight command software.) The effort authorized for the functional specification stage for purely prototyping activities represents 42 man-months.

(2) I cannot put a number on the relationship between the effort of developing prototyping software and of developing operational software, because the real software was developed from specifications in different machinery by different companies. From all accounts, this ratio is well below one.

G.Bouche, GE

(1) What are the key differences between the prototype software and the operational (embedded) software that make it so much easier and faster to write the prototype software, although it will be fully functional?

(2) What is the approximate difference in man-years between the prototype software and the embedded software in RAFALE?

Author's Reply

(1) The prototype software was developed very rapidly on nonspecific materials and under different constraints from the operational software — functional reliability, efficacy, language, respect of real time, etc. — remaining functionally identical nonetheless (from the close temporal aspect). Only the test trials resulting from the prototype are compatible after formatting of real interfaces.

(2) See answer to the question posed by Mr E.Cambise.

The Avionics Software Architecture Impact on System Architecture

C. Douglass Locke, PH.D.
Senior Programmer
M/D 0210
IBM FSD
Owego, NY 13827

Abstract

Existing avionics systems are designed as a set of subsystems integrated by command and control software in a central processor system. In future avionics systems, the complexity of this critical software is expected to increase dramatically, leading to potentially explosive growth in both software costs and the likelihood of critical software errors. As the software increases in both its complexity and its criticality to the success of the overall mission, (e.g., by the infusion of AI, image processing, distributed processing, etc.) the vulnerability of the system to software errors also increases dramatically.

This problem can, and, we feel, must, be alleviated by the use of new methods for defining the system architecture, and allowing the software architecture to constrain the design space of the hardware/physical architecture. Thus, the process of developing the software architecture must change both in its development methods, and the avionics system design cycle stage at which it is performed. *It is thus critical for the software architecture to be a driver of the system architecture design decisions.*

In this paper, we consider the technology developments which have led us to this problem and their impact on the functionality and design of new systems. Following this, we discuss the current sequence for performing the physical and software architectural design, including a definition of the software architecture, with examples. From this, we discuss what we feel are the likely consequences of using these methods for designing such new avionics systems, and one potential solution to these problems.

1.0 Introduction

It will come as no surprise to practitioners in avionics systems that the characteristics of such systems have undergone a rapid evolution in the last 10-15 years, and that a similar, if not greater rate of change is ahead in about the same time interval. The nature of these changes may be characterized using a variety of metrics, some of which are summarized in Figure 1. For those characteristics which can be numerically evaluated, the growth is described by a geometrical progression, not by a linear progression. During this period, the sequence of steps by which an avionics system is designed has not undergone such a rapid change; a question which we would like to address is, "will our current system design methods be adequate for the types of future systems characterized by these metrics?"

	Past	Present	Future
Human Interface	Manual	Man-in-the-Loop	Automated
I/O Data Rates	10*2/sec	10*4/sec	10*6/sec
DP H/W MIPS/Memory	0.4/64K	2/256K	10/4M
No. of Processors	1	2-4	10-20
S/W Processing Types	Math. Equations	Command & Control	AI, Sensor Fusion, Video
Primary Constraint	H/W	H/W + S/W	S/W

Avionics System Evolution

Figure 1

In the past, avionics systems were simple by modern standards, with architecturally small, single computers with processing limited to conceptually simple algorithms such as vector rotations, ballistics computations, display scaling, manual operator interfaces, and with overall functionality constrained primarily by the amount of hardware supportable by the platform in terms of weight, power, and volume. In current modern systems, the avionics system supports man-in-the-loop operator interfaces, requiring more powerful and larger processors capable of command and control processing in conjunction with the human decision maker. Such system designs are constrained both by the hardware, as previously, and the software which must be able to use the constrained hardware to provide the required functionality. In future systems, characterized by automated (with manual override) control with its inherent need for increased fault tolerance, AI, sensor fusion functions, and rather powerful computers, the primary system design constraint will be the software (i.e., the set of algorithms and data structures required to fulfill the system specifications).

These future systems will require changes to the methods by which the system design is performed; the geometrical changes in the system complexity leading to greater numbers of much more powerful processors will not allow us to continue designing systems in the same ways as before.

The process of designing an avionics system may be described as a sequence of design stages or steps. Although the number of steps and their exact nature is highly variable, for the purposes of this paper we identify four top-level steps in Figure 2.

This sequence is, of course, predicated on the assumption that these stages may be sequentially performed; i.e., that the information needed to initiate each stage originates only with the preceding stages. For past and current systems, this assumption is justified, but it is the premise of this paper that this assumption may not be warranted in highly complex systems such as some of those currently on the drawing board, particularly for the last two steps listed in Figure 2. It is not obvious, in systems with a large number of interconnected, powerful processors, that it necessarily will be possible to design the processes for each processor without subsequently modifying the number of processors and their interconnections, thus significantly increasing the cost and risk of the avionics development.

(1) Conceptual Design -- Determining the concepts to be used in the avionics design. This activity derives these concepts from the system specification; for example, when the system specification calls for an aircraft to use acoustics for underwater detection, the conceptual design will identify the type of acoustic devices required (e.g., sonobuoys, dipping sonor) and structures, interfaces, and controls required.
(2) Functional Design -- Given that the set of system concepts has been defined, the actual functions to be performed in support of each concept must be determined. An example of this would be the decision to deploy sonobuoys automatically in response to passing a given geographical position.
(3) Systems Architecture -- Having determined a set of functions to be provided, the systems architecture can be defined. Examples of systems architecture decisions are the devices required to handle a sonobuoy system, the nature of their interconnection, and the specific operator actions which are needed to manage these devices. Included in the current systems architecture phase are decisions about the number and type of processors needed and the nature of their interconnections.
(4) Software Architecture -- Upon presentation of a system architecture, the software architecture may be determined. This activity consists of such actions as designing the functions to be provided by each processor, determining the communications protocols to be used, defining what, if any, operating system should be used (including how the specific operating systems constructs are to be used), designing the modular breakout of the software, etc.

Conventional Avionics System Design Sequence

Figure 2

Considering, for example, the LAMPS Mark III system, we note that the LAMPS conceptual development grew out of experience with previous LAMPS systems, as well as out of experience with several relatively simple prototypes constructed and tested over several years. For the full scale development, these concepts were used to design an avionics system consisting of 5 general purpose computers, using a variety of interfaces (point-to-point channel attachments, data link, and a communications bus). The software architecture was then determined independently for each of these computers, along with the communications protocols needed to fulfill these requirements. For the system/software architecture steps, this assumption of independence was valid for two principal reasons:

- The LAMPS problem, in spite of the presence of a large number of functions, was very simple structurally.

- Because of LAMPS' inherent structural simplicity, a federated distribution of functions among the various processing elements provided a satisfactory solution to the software architecture (i.e., a federated distribution is one in which the allocation of functions among processing elements is physically fixed because of asymmetrical processing capability and interfaces). A by-product of such a federated functional distribution is that the choice of communications protocols for each communication path is independent of any other communication paths.

It is clear that the avionics systems of the future, moving in the directions of increasing function exemplified by ATF and LHX and illustrated in Figure 1, are at least an order of magnitude more complex, requiring more fully integrated functions. From the point of view from which this paper is being written, the key difference between current systems such as LAMPS and such future systems lies in the degree of interconnection among the various subsystems. For example, in LAMPS, the sensor subsystems, such as the acoustics, have no direct interface with the navigation subsystem other than through a common data base. On the other hand, for systems such as ATF, virtually all functions are envisioned to be fully integrated; e.g., the FLIR system images can be processed both for targeting information to be coupled to weapons delivery, and for navigation fixing. Beyond such direct interfaces, the FLIR video must be digitally enhanced for pilot presentation, involving both extensive hardware and software interfaces.

It is important to note also that the sequence of decisions in the design process is influenced by the factors which drive the design. In the past, hardware physical characteristics such as weight, power, and size, as well as performance, were prime factors in determining the physical design; thus making the hardware choices first, followed by the software architecture within the physical constraints made sense. It is the premise of this paper that the current avionics system design methodology applied to

future complex systems will lead to expensive (cost and schedule) development iterations and suboptimal implementations. Now, in the face of the increased system complexity required, the difficulty in successfully integrating the required software functions in these systems forces the software architecture to drive the overall physical design although physical attributes such as weight and power will, of necessity, constrain the resulting design.

In the remainder of this paper, we discuss the relationship between the system architecture and software architecture in Section 2.0, a new approach to a coordinated system/software architecture in Section 3.0, and identify the key conclusions in Section 4.0.

2.0 System Architecture vs. Software Architecture

The process of defining an architecture consists of defining a structure and the relationships between the entities which make up the structure. This definition of architecture applies to many fields; for example, in construction, the architecture consists of defining the relationship between shapes, colors, textures, and functions. Similarly, in avionics systems, the process of defining the avionics architecture (i.e., the system architecture) consists of defining the relationships between entities such as sensors, actuators, and weapon systems and the human(s) who will operate and use the system.

For discussion, we choose to separate the concept of the avionics system architecture into the physical architecture and the software architecture. The physical architecture consists of the structures and relationships among the physical elements of the system, e.g., the processors, peripherals, busses, sensors, and actuators. The software architecture, on the other hand, consists of the assignment of the control functions which actually perform the coordination of the system entities into computational elements, and the relationships between the computational elements and the data structures. In the software architecture, the objects consist of two major categories: 1) the processes which perform the functions required of each computational element, and 2) the data structures containing the information to be processed by each of these processes. To this point, the software architecture can be thought of as if we were targeting to a uniprocessor. The software architecture also then includes the partitioning of these processes and data structures into a set of physical processors. Thus, the software architecture is developed in conjunction with the system architecture to define the number and type of the processors, but the software architecture alone determines the placement of the process functions among the processors.

In current systems, the software architecture could successfully be performed sequentially following the system architecture because there are only a small number and type of processing elements, and the function of each process was very closely connected with the system architectural elements themselves. Thus, for example, in the LAMPS helicopter, we find the primary command and control computer to be a single entity, with no backup, coordinating with a set of peripherals and processors, which were, in general, not general purpose computers. Specifically in the LAMPS helicopter the other principal processors consist of a sensor processor for performing acoustic analysis, and an ESM processor for analysing incoming electromagnetic radiation. The functions performed by these processors is specific to these subsystems, and no functional backups on the aircraft were felt to be required (or feasible at that time) across the system boundaries. Thus, the allocation of function within the LAMPS helicopter to the various processors was dictated by the choice of sensors themselves.

In modern avionics systems, on the other hand, we find that the processing requirements are sufficiently great that it is necessary to have a number of processors in each of the avionics functional elements. This means that the processor-function assignment is both more flexible and more complicated, as issues of communications protocols, fault tolerance, and reconfiguration become critical to meeting the overall avionics systems objectives. If decisions relating to the software architecture are left until the processors and their interconnection have been defined, the design space for the software architecture will likely have been so constrained that it will not be feasible to fulfill the overall system objectives.

3.0 An Approach to System/Software Architecture for Complex Systems

If the generation of a system architecture followed by the software architecture seems not to meet the needs of future system designs, the logical question to ask is "What sequence of system/software architectural development will result in both a reasonable architecture and a working system in a reasonable period of time?" In Figure 3, we show a refinement of steps 3 and 4 for Figure 1 into 7 steps. These 7 steps are contrasted with a new sequence of seven steps (shown in Figure 4) to address the projected highly complex environment of the future.

In order to evaluate these seven steps and their use in constructing a real-time avionics system, let us begin by considering the steps as they are now generally performed (see Figure 3). It will be observed that there are some fundamental dissimilarities between the sets of steps in Figures 3 and 4. In the proposed approach we are considering, we find that by the time step (3) is completed, the high level algorithms and data structures are already in place that are required for the overall system solution. We note that these are not the low level algorithms (sometimes called "math flows" in the past) which describe the processing on behalf of individual sensors or actuators. For example, we would not expect to find here the details of the navigation equations used to generate current position from an inertial navigation system, but we would expect to find the flow of information from the navigation system to the low-light level TV or the radar, ECM equipment, etc. The key here is that we have defined the algorithms and data structures based only on the sensors and actuators in use, as well as the human interfaces required, without having yet considered the information processing or communications structure. In other words, we consider the processor and communications structure decisions to be a result of, not the determiner of, the software architecture, from the top level perspective.

Therefore, by the time we have completed step (4), and prior to defining the processor and communications structure, we have determined the complete functionality at the system level, and have defined the flow of information that must occur between all of the pieces of equipment that actually interface with the operational environment. By contrast, we note that this is not accomplished until after the definition of the data processing equipment and communications equipment in the conventional avionics system software engineering architecture procedures.

(1) Define each of the sensors and actuators (including weapon systems) for the system, based on the top level system requirements. This process accounts for required vehicle performance, flight envelopes, weight, etc., but disregards information processing and communications requirements.

(2) Define the human interface functions. This consists of a list of operator functions, both input and output, including a model of the required operator responses and estimate of required operator precision and speed.

(3) Define the computational elements and communication structures to be used to control these sensors and actuators and interface with the human operators. Consider the fault tolerance, fault containment, and degraded modes.

(4) Define the algorithms and data structures for each of the sensors and actuators defined, constraining them to fit with the computational and communications equipment.

(5) Deduce the information flows among each of these components for all required system modes. This includes the data structures to be communicated, the rates at which communications must take place, and the required precision of each data element.

(6) Analyse the resulting system architecture for communications or processing bottlenecks, probably using a combination of analytical and simulation techniques.

(7) Iterate steps 3. through 6. until the system reliability requirements have been satisfied.

Decomposition of Conventional System Architecture Development Sequence

Figure 3

(1) Define each of the sensors and actuators (including weapon systems) for the system, based on the top level system requirements. This process accounts for required vehicle performance, flight envelopes, weight, etc., but disregards information processing and communications requirements.

(2) Define the human interface functions. This consists of a list of operator functions, both input and output, including a model of the required operator responses and estimate of required operator precision and speed.

(3) Define the algorithms (processes) and data structures for each of the components (e.g., sensors and actuators) defined in step 1. These algorithms define the interrelationships between these components, not the computations performed internally. For example, this might define how information from a navigational component would be used to perform a flight maneuver or position a weapon, not how accelerations measured by an inertial navigation system are converted to velocity or position.

(4) Deduce the information flow among each of these components for all required system modes. This includes the data structures to be communicated, the rates at which communications must take place, and the required precision of each data element.

(5) Partition this information flow, considering requirements for fault tolerance, fault containment, and degraded modes (including potential software faults in addition to hardware faults) into processing elements and supporting communication structures (e.g., busses).

(6) Analyse the resulting system architecture for communications or processing bottlenecks, probably using a combination of analytical and simulation techniques.

(7) Iterate steps 5. and 6. until the system reliability requirements have been satisfied.

Proposed System Architecture Development Sequence

Figure 4

The big difference is that in the event of difficulty meeting any of the data processing constraints, such as fault tolerance, fault containment, and degraded modes, in the normal procedures the entire data processing structure and communications structure may have to to be modified in several design iterations. This cannot, of course, be done until each problem has been diagnosed, which may frequently not occur until after the system has been developed and partially implemented.

This risk of needing major design iterations is a critical failing of such a system since it leads to high cost and/or poor quality, and we believe that the risk can be contained with our new approach. This belief is supported by the now well-documented fact that the early detection of errors in a system implementation dramatically reduces the cost to correct them. We note that the iterative part of the proposed approach affects only the last two steps, steps (5) and (6); rather than affecting the whole design, as in the conventional approach.

As an example of a design problem which can be avoided by this approach, there are a number of cases in which a set of interconnected computers has been configured to provide a high degree of fault tolerance by incorporating suitable redundancy. It is frequently the case, however, that when the software is being designed, the required functions could not be implemented without additional hardware support because the underlying algorithms and data structures had not been designed and considered prior to defining the interconnection structure.

It might be argued that the current approach (see Figure 3) allows the detailed hardware design to begin earlier in the system design cycle. This is true, and in the past, an early start on the hardware design was considered critical to meeting overall system schedules. However, the new (correct) emphasis is on the use of off-the-shelf hardware elements, reducing the need for allowing for long lead times for hardware design, and the increased software complexity has caused it to increasingly dominate both the system cost and schedule.

From the point of view of the skills to be used for developing a system under this paradigm, there is another fundamental difference between these approaches. In steps (3) and (4), at the very beginning of the high level system design, the software engineer must be already involved. The skills needed for this step consist of a knowledge of the data processing and communications issues of the avionics system, rather than the systems issues which were needed in steps (1) and (2), and which will be needed in steps (5) and (6). To an increased extent, system design decisions will be determined by software considerations. While it is possible that a few individuals have the requisite education and experience to perform all of these steps, the normal case is that personnel with knowledge of such things as the flight dynamics, mission profiles, and operator interface will not be able to optimally define the information processing and communications design. Thus, this approach encourages the use of system architecture teams with greater software skills and awareness to define the high level design.

4.0 Conclusions

The current sequential nature of the system engineering, determining a physical architecture followed by performing the software engineering to produce the new, extremely complex avionics systems makes the design process prone to sub-optimal solutions or unworkable designs leading to critical functional errors. These errors are not so much the traditional types of errors exemplified by targeting errors or incorrect displays, but are likely to be some of the more subtle problems such as failure to meet timing requirements (appearing usually as "intermittent" errors), or, worse yet, failure to respond properly to component failure with an appropriate reconfiguration or degraded mode.

We have described a proposed new sequence for performing such a major avionics system design by intermixing the system/software engineering tasks to isolate the complexity introduced at each step, and minimizing the number of factors which must be considered in the iterative part of the design process. This should result both in a less error-prone high-level design, and in a process with a much greater likelihood of success at a more controllable cost. IBM is strongly committed to the production of such systems, and is currently carefully considering approaches such as the one presented in this paper to ensure that the complex avionics systems needed for the future can be as successful as those produced to date.

A COMPARISON OF INTEGRATED AND SEPARATE SYSTEMS FOR FLIGHT CONTROL AND NAVIGATION

by

H.Buitkamp
LITEF (Litton Technische Werke) der Hellige GmbH
Lörracher Straße
D 7800 Freiburg
Germany

1. Summary

Modern aircraft use signals from gyros and accelerometers in various subsystems, among these flight control, attitude/heading and navigation systems. Utilisation of the same set of instruments for these three functions could be envisaged.
This investigation discusses some possible steps leading to an integration of navigation and flight control functions into one system. Obvious advantages of such a system could be a decrease in weight, power consumption and volume.
Emphasis of this paper is laid on technical feasibility and on possible development risks associated with the application involving inherently unstable aircraft.
The main requirements for reliability, accuracy, and bandwidth of the sensor data differ according to whether the application is for navigation or for flight control. A study of these requirements forms a reasonable basis for the concept of an integrated system.
The conclusion is that an integrated system is appropriate for an inherently stable aircraft, but that a lot of risks are involved in the development for control configured vehicles.
Investigations have shown that the essential reduction in weight, power, and volume for systems combining navigation and flight control tasks is very difficult to achieve.
Consequently, first-integration-stage systems, which include only the flight control sensors and the attitude/heading determination, are recommended. Such systems are already produced at LITEF and are an integral part of the EAP[1] which involves an inherently unstable aircraft. During the first flight trials in August 1986 the system successfully demonstrated its capabilities.

2. Introduction

Modern fighter aircraft require an appropriate FCS[2] to provide sufficient handling quality and adequate stability in order to reduce the pilot's workload and simultaneously provide a stable weapon delivery platform. In addition an autonomous navigation system and an AHRS[3] are required in most aircraft applications.
To date the inertial data required by the individual functions - FCS, AHRS and INS[4] - has been provided by independent sets of gyros and accelerometers. The redundancy of the sensors increases the weight, size and cost of the whole aircraft. Furthermore an increase in the number of sensors introduces extra maintenance problems. Therefore the utilisation of the same sensors for flight control purposes, for attitude and heading determination and for navigation purposes could be envisaged. An integrated system has to deliver the inertial sensor data with at least the same accuracy, functional reliability and lower life cycle costs as that of a separate system.
Using strap down technology, integrated systems for flight control, attitude/heading determination and navigation purposes have been successfully implemented in the transport aircraft field. This goal also seems to be achievable in the very near future for agile inherently unstable fighter aircraft, but there is a great deal of work required.

A brief summary of recent development efforts in this field forms a basis for our own investigations.

---

[1] Experimental Aircraft Program

[2] Flight Control System

[3] Attitude and Heading Reference System

[4] Inertial Navigation System

2.1. History

One of the first integrated systems was built at Litton in 1976. It contains four DTG's[5] and eight accelerometers in a pyramidal axes configuration. It consists of four boxes and has a weight of 27.7 kg. The system is not applicable for flight control and navigation of a highly agile CCV[6] fighter aircraft due to the time delays, the measurement ranges and the navigation accuracy.

Another important milestone in the development of an integrated system was the MFCRS[7] program, which was started in 1980 by the USAF together with McDonnell Douglas. The major components used are two skewed mounted inertial navigation systems H 421 manufactured by Honeywell and installed in a F-15 Eagle. The idea behind the skewed mounting of the two orthogonal strap down systems was to provide sufficient redundancy for flight safety critical information by using two accurate, costly and heavy inertial systems only. It was discovered that the location of the two boxes in the aircraft and the noise of the compensated sensor data have a strong influence on the FCS. This affects the handling quality of the aircraft although the navigation accuracies are hardly reduced. It became further apparent that the latency of flight safety critical data provided by the inertial systems allows only flying of the aircraft under conditions where dynamics are not critical. Further investigation should be carried out in order to determine if the navigation system, with only a few modifications, could be used for flight control.

In 1983 Litton began development of the IISA[8]. The system consists mainly of two similar boxes, each with three RLG's[9] (28 cm) and three accelerometers. The boxes are to be installed back to back in the aircraft, thus forming a hexagonal sensor configuration.

There arose, during the development of IISA, several questions and solutions associated with a common sensor block for flight control and navigation. An indication has also been attained as to the necessary power consumption, weight, volume and cost of an integrated system as compared to the solution using individually designed systems for each application. The first flight tests are planned for 1987.

The IISA has been conceived for inherently stable aircraft. If used for inherently unstable aircraft, it would be necessary to reduce the time delay and increase the bandwidth of the flight safety critical control data. To build such a system, using state of the art techniques, would involve considerable development effort along with the associated risk and cost.

3. Requirements for Inertial Systems

The choice of sensors in inertial systems combining the multiple functions of FCS, AHRS and INS into one integrated system, is dictated by the function having the highest requirements. In general all sensors should have a high accuracy, integrity and a long life span.

The requirements imposed on the inertial sensors and the safety critical data differ depending upon whether they are used for a FCS, an AHRS or an INS.

A table with the most important sensor data requirements is shown below.

Function	FCS	AHRS	INS
Bandwidth	> 50 Hz	20 Hz	< 20 Hz
Data latency	< 10 ms	not critical	not critical
Reliability	safety critical	safety critical	mission critical
Redundancy	quadruplex	duplex	simplex
Accuracy, rel	5000 ppm	500 ppm	50 ppm
Accuracy, abs	100 deg/h 10 mg	1 deg/h 1 mg	0.01 deg/h 0.1 mg

Table 3-1 Sensor Data Requirements

As can be seen from the table the bandwidth of the safety critical data for flight control should be greater than 50 Hz. This value is calculated assuming a 6 Hz bandwidth for the aircraft dynamics of a CCV and the fact that the bandwidth of the flight control data should exceed the bandwidth of the aircraft motion by a factor of eight. The bandwidths quoted for an AHRS and an INS are only rough mean values and actually vary with different aircraft.

---

[5] Dynamically Tuned Gyro

[6] Control Configured Vehicle

[7] Multifunction Flight Control Reference System

[8] Integrated Inertial Sensor Assembly

[9] Ring Laser Gyro

Whereas a time delay for the AHRS and INS functions is not critical, a time delay greater than 10 ms for flight control data could cause a destabilising effect in the aircraft control system.
The reliability can be categorised into "safety critical" and "mission critical". The former category applies to flight control data, the loss of which would cause the aircraft to be no longer controllable. Under special circumstances the three attitude angles could also be safety critical. The loss of an INS does not influence the stability of the aircraft, but loss of attitude information make it impossible to complete the mission.
Usually a CCV aircraft needs a quadruple redundancy for the flight safety critical data. The attitude angles are required duplex delivered by the INS and by a SAHRS[10]. Cost and weight restrictions allow the use of only one INS in most fighter aircraft applications. The relative accuracy of the sensor data of an INS with medium accuracy should not exceed 50 ppm including all error compensations. The same requirement for an AHRS is 10 times lower. It is sufficient that the flight control data have a relative accuracy of 0.5 percent. The accuracy in the determination of angular rates and accelerations differ in the three applications by a factor of 100 and 10 respectively.

In a nutshell, the task is to integrate the diverging requirements of high bandwidth and highly reliable data without special emphasis on accuracy for flight control
and
highly accurate data without special emphasis on latency, noise and bandwidth for navigation.

4. Steps of Integration

There are at least three ways of solving the flight control, the attitude/heading and the navigation tasks for a modern CCV, bearing in mind that quadruplex redundancy for flight safety critical data and duplex redundancy for both attitude/heading and navigation functions are required.

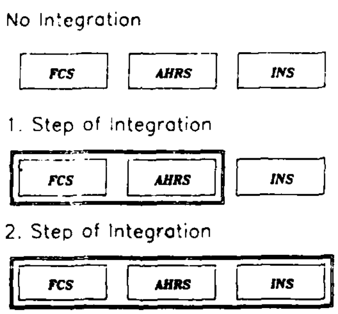


Figure 4-1 Steps of Integration

When using nonintegrated separate sensor blocks, it is necessary to employ four independent channels for flight control purposes and a duplex sensor package for both attitude/heading and navigation tasks. This leads to a maximal configuration and to the use of plurality of sensors for rates and accelerations.
One configuration described below is to use an EAP/AMSU[11]-system together with one INS. This AMSU-system is the result of the first step of integration outlined in figure 4-1. It provides flight control data for stability augmentation and the autopilot function of an inherently unstable aircraft, and redundant attitude and back-up-navigation information. This system is already an integral part of the EAP-FCS and has successfully completed flight trials.

In the next development step the flight control sensors, the attitude/heading determination and the navigation functions are combined into one system. The principle procedure for developing such a system and the arising risks will be discussed in chapter 4.2.

---

10 Secondary Attitude and Heading Determination System

11 Aircraft Motion Sensor Unit

4.1. Combination of Flight Control and AHRS Functions

As already indicated in the introduction, there are at least two approaches to the integration of inertial data into an airborne weapon system. The AMSU approach combines the determination of angular rate and linear displacement with the provision of attitude and heading Euler angles.

The AMSU-system, the main features of which are shown in table 4-1, has been developed especially to suit the requirements of the EAP-FCS. It provides high bandwidth, fresh and highly reliable low noise data in order to achieve good handling quality and stability of the aircraft. In addition, the system provides attitude, heading, inertial vertical speed and inertial altitude.

mass	39.2 kg	(4 boxes)
volume	38.0 l	(4 boxes)
redundancy	quadruplex	
range	250°/s	roll rate
	125°/s	yaw rate
	85°/s	pitch rate
	12g	accelerations
output	512 Hz	rates, accelerations
	42 Hz	attitude, heading
bandwidth	45 Hz	rates, accelerations
staleness	2,5 ms	rates, accelerations excluding filters
availability	$9 \cdot 10^{-7}$	probability of loss of aircraft due to AMSU/FCC
	$2 \cdot 10^{-6}$	probability of loss of aircraft due to FCS

Table 4-1 Main Features of the EAP/AMSU-System

The function as a sensor for flight control data and attitude/heading determination imposes a particular structure on the AMSU system. The pulse counts delivered from the gyros and accelerometers are scaled, compensated and output at a frequency of 512 Hz by a fast TMS 320 processor. Two MC 68000 microprocessors in parallel compute the attitude and heading equations along with internal built-in-tests. These tasks are carried out at a frequency of 42 Hz. The initial heading is obtained by autonomous gyrocompassing. The interconnection of the AMSU with the flight control computer uses a dedicated serial digital link. The interconnection of the AMSU-system with the quadruplex flight control computer is shown in figure 4-2 below.

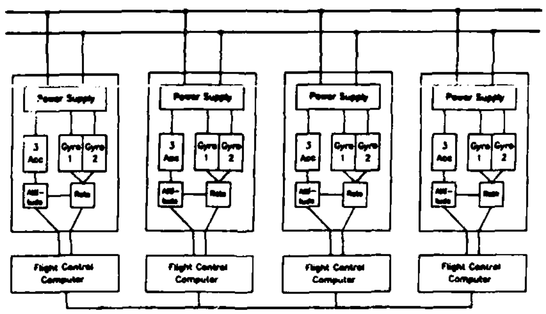


Figure 4-2 Interconnection EAP/AMSU - Flight Control Computer

The roll, pitch and yaw rates are measured in each box. In addition each box contains three dry, force rebalanced B-280 accelerometers in an orthogonal configuration.

In order to suit the high dynamics of a CCV, the measurement ranges for rates and acceleration are respectively set to max. 250°/s and 12 g. The bandwidth capability and data latency of the angular rates and accelerations are better than 45 Hz and 2,5 ms respectively, excluding aircraft structural filter and anti-aliasing filter transfer functions.

As integrity is very important for flight safety critical equipment it should be recalled that the MTBF of the LTR-81[12] which incorporates two K-273 gyros and three B-280 accelerometers exceeds 10,000 hours proven within more than 400,000 equipment flight hours in the commercial transport aircraft environment. Under the above mentioned conditions this gyro has shown a MTBF of more than 100.000 h. To our knowledge these figures are presently higher than the equivalent number experienced with ARINC 705 RLG-IRS[13] in the same environment.

---

[12] LITTON Transport Reference 1981

[13] Ring Laser Gyro-Inertial Reference System

The effort required to adapt an AMSU-system for EFA[14] is mainly to reduce weight and to improve integrity from the EAP/AMSU. In parallel with these improvements the complexity can be reduced by accounting for the fact that duplex redundant Euler angles are sufficient and twelve accelerometers would not be required. The goal to reduce the weight can be achieved by combining two completely independent angular rate channels with one attitude channel in one housing. This will lead to a two box solution including a quadruplex angular rate and acceleration output and duplex attitude/heading output. A housing concept of 3/4 ATR size is shown in figure 4-3 below.

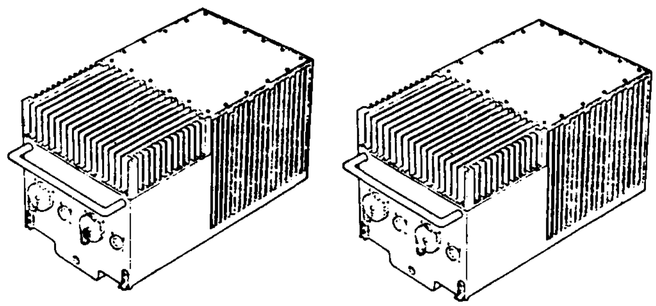


Figure 4-3 EFA/AMSU System

4.2. Combination of Flight Control and Navigation Functions

Any development of an integrated system for flight control, attitude/heading determination and navigation functions has to consider the impact on safety and operation of the whole aircraft. For a CCV application it is typically required to have available quadruplex redundancy for the stability augmentation data, duplex redundant attitude and simplex automomous navigation data.
It is necessary to use strap down technology, as opposed to a platform system, because the latter does not deliver sufficiently fresh stability augmentation data due to the necessity of differentiating the Euler angles and subsequent filtering of the noise generated.
In order to reduce vulnerability it is advisable to split the system into two boxes, each of which is designed to deliver both navigation and flight safety critical data.
For non-augmented medium accuracy INS's laser gyros, of weight 1.2 kg and an accuracy better than 0.01°/h corresponding to a path length of 25 cm, are required.

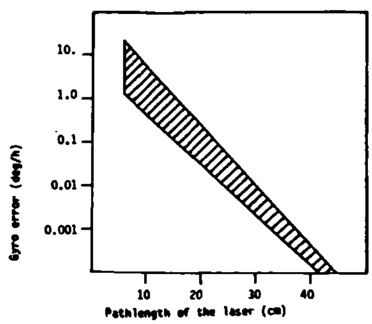


Figure 4-4 Errors of Laser Gyros

---

[14] European Fighter Aircraft

To obtain the demanded quadruple redundancy using orthogonal measurement axes, 4 x 3 = 12 RLG's would be required. It can be shown however that six skewed mounted RLG's, three in each box orthogonal to one another, are sufficient for the necessary redundancy.

During the development stages on an integrated system the following points have to be considered:

- Evidence of integrity should be provided.
- The INS should have at least three RLG's.
- The system should be repackaged in such a way that the safety and redundancy requirements are satisfied.
- The sensor block should be rotated so that it is skewed in relation to the housing.
- Independent self-contained electronics for each flight safety critical rate channel should be provided.
- The individual channels should be mechanically isolated by walls.
- The safety critical software and software-tests should be developed to appropriate standards.
- An appropriate redundancy management and BIT[15] is required.
- The filters, time delays and accuracies, which depend on the performance of the aircraft should be discussed with the people responsible for flight control.
- A new flight control computer should be developed (if integrated).
- An appropriate location for the system in the aircraft should be established.

4.3. Main Problem Areas

The main problem and risk areas arising during the development of the integrated system are:

- Vibration isolators
- Data processing
- Redundancy management
- Separation of the system

These points are discussed in detail below.

4.3.1. Vibration Isolators

There are generally three kinds of translational vibrations specified, namely:

- vibrations induced by aerodynamical effects
- vibrations induced by engines
- vibrations induced by gunblast

The specified spectra, especially in low frequency ranges, occur seldom in practice. In reality the vibration spectra have RMS[16]-values which are approximately ten times less. It is left up to the customer to decide whether the spectra are used for a realistic simulation environment or only to test the system under extreme conditions.

When choosing the vibration isolators and the sensor block mounting method it is necessary to take into account the bandwidth of the flight dynamics of the aircraft.
A high bandwidth requires high natural frequencies of the vibration isolators and a corresponding hard mounting of the sensor block. The vibration power at the sensor block increases with the natural frequency of the vibration isolators so that the burden on the sensors, along with the resulting errors, rise. It can be said that the sensor errors and the subsequent navigation accuracies are almost proportional to the square of the vibration power. Due to this fact it is necessary to use soft vibration isolators for appropriate navigation accuracy.
Typical frequencies are shown in the following diagram.

---

15 Built-In-Test

16 Root Mean Square

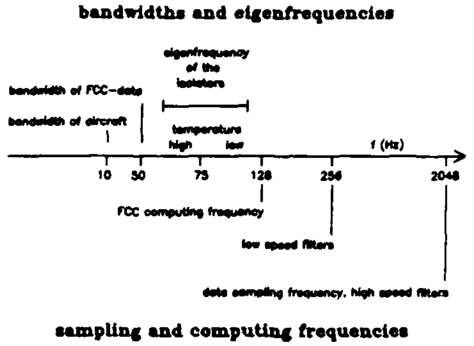


Figure 4-5 Frequencies for Flight Control Purposes

Isolators corresponding to inertial navigation systems have natural frequencies of around 20 to 40 Hz at 20°C and a transmissibility of about 4 to 6. For flight control purposes however, a value almost twice as high is required depending on the high bandwidth (6 Hz) of modern inherently unstable aircraft. This value can vary with low temperatures and high vibration levels by a factor of 1.5 up to 120 Hz. The effect of this is to decrease the accuracy of the navigation by a factor of about 4.
It can be concluded that the choice of the vibration isolators depends on the one hand on the high dynamics of the aircraft and on the other hand on the soft mounting of the sensor block necessary to obtain the required navigation accuracy.

In order to reduce the temperature-dependent variation of the natural frequency it is useful to define an appropriate temperature range. The choice of the elastomeres for the isolators can be made optimal by restricting the temperature range. This reduces the variation in the natural frequency and transmissibility of the isolators, as shown in figure 4-6. In general a small variation in the eigenfrequency can be achieved at the cost of a high variation of the transmissibility and vice versa. These problems can be alleviated however by either limiting the temperature range or by installing a sensor block heater, if the application allows.

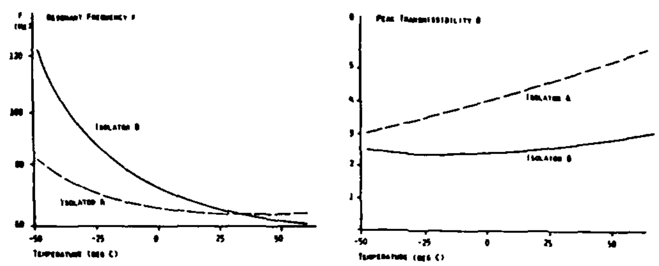


Figure 4-6 Variation of the Natural Frequency and Transmissibility with Temperature

When mounting the sensor block it is very important to consider that the rotational and translational natural frequencies thereof are wide apart. The reason for this is that the coupling of the translational into the rotational vibrations decreases with the increasing difference of their eigenfrequencies. Rotational vibrations of the sensor block will cause the angular rates of the aircraft to be incorrectly determined.
A non-isoelastic mounting of the sensor block causes transformation of translational into rotational vibrations and produces dither induced coning motion. An optimal mounting for a cubic sensor block can be achieved using four isolators forming a tetrahedron, if the space available allows. This solution will produce a ratio between the rotational and translational eigenfrequencies, approaching the maximum attainable value of two.

If the static and dynamic mass-unbalance of the sensor block is known, then it is possible to determine the rotation of the sensor block from the specified translational vibrations by use of the transfer function. By this method it is possible to compute system errors like coning and sculling and their effect on navigation accuracy.

Another important aspect to be considered is the effect of the hard isolators on the dither behaviour of the laser gyro. The laser gyro is connected via a dither spring of high rotational transmissibility (300) to the sensor block, which in turn is mounted on isolators, as shown in figure 4-7. It can be shown using the theory of coupled oscillators that the effective transmissibility of the dither spring depends on the natural frequency of the isolators.

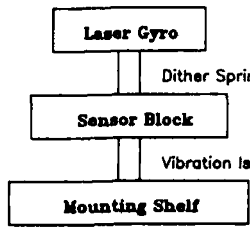


Figure 4-7 Mounting of the Laser Gyro Block

If the effective Q of the dither spring falls to a value in the region of 150 then the isolators will absorb too much energy from the dither driver and furthermore the excitation of the actual glass block of the RLG will become so weak that the lock-in threshold cannot be overcome. This limits the frequencies of the vibration isolators. Figure 4-8 shows how the effective Q decreases with the increasing resonant frequency of the isolator.

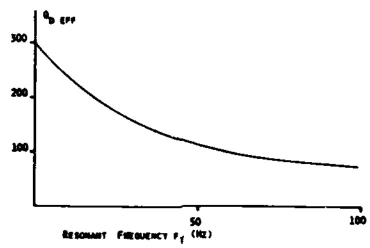


Figure 4-8 Variation of the Transmissibility of the Dither Spring

4.3.2. Data Processing

An important problem area in the integration of navigation and flight control systems is the definition of the required flight control filters and their time delays. The time delay is not much of a problem for attitude/heading determination and for navigation purposes.
The filters are divided into two categories - high speed and low speed. The high speed filters have to reduce the dither and quantisation-induced gyro noise. The computational frequency of these filters has to be identical with the sampling frequency of the data (2048 Hz). These filters typically produce a time delay of about 5 ms. Sensor error compensation, along with the transformation to orthogonal measurements, reasonableness tests and BIT, produce a further 2.5 ms time delay. The resulting data are finally compensated for structural modes, isolators, autopilot characteristics, etc. using the low speed filters. Typical transfer functions and sensor noise characteristics are shown in figure 4-9.

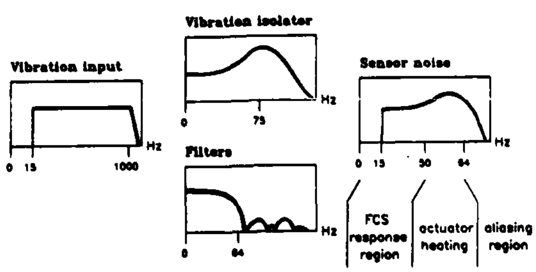


Figure 4-9 Transfer Functions and Sensor Noise

The time delay associated with those filters mainly depends on the algorithms used and is approximately 5 ms including the data transfer to the flight control computer. This yields a total time delay of 12.5 ms which corresponds to an excess phase of 45° at 10 Hz.

When using skewed sensor axes it is important to ensure that the measurements are made simultaneously. If this is not the case then the transformation to aircraft axes frame will lead to grave errors in the flight control domain. A typical value for the time synchronisation is 0.02 ms. A possible solution to this problem is high speed data transfer or processor synchronisation. If connection wires for processor synchronisation are used then this could lead to single point failures which of course are to be avoided.

4.3.3. Redundancy Management

An important point in the design of an integrated system is the development of an efficient RM[17]. In principle the RM is carried out in three stages:

The first step is the failure detection procedure, which distinguishes between hard and soft failures. Hard failures have to be detected within 50 ms via BIT procedure, whereas the soft failure detection can be somewhat longer.

The second step is the isolation of the failure using respectively BIT and parity equations for hard and soft failures. The choice of the thresholds necessary for the parity equations depends on the required accuracy of the flight control data, on the noise levels of the sensors and on the dynamics of the aircraft. In order to avoid false alarm it is necessary to take into account structural bending when using a system with two boxes separated in the aircraft.

The third step is the reconfiguration of the system after an arising failure. This is achieved by design equations which transform the intact skewed measurements into orthogonal measurements taking into account, however, any switching transients.

The detection and isolation of the failures and the reconfiguration of the system introduce a time delay which has to be added to the existing failure detection time delays.

One of the most important points is to consider the single point and common mode failures which subsequently lead to the loss of the whole aircraft. Some of these failures due to the sensor package for flight control are listed below:

- broken isolator
- open latch in the mounts
- wrong software (common mode failure)
- defect busses
- etc.

4.3.4. Separation of the System

Vulnerability makes it necessary to split the system into two boxes. This leads to sensor location problems in the aircraft. A solution to this problem could be to search for an optimal location in the aircraft which depends on the fuselage bending nodes and antinodes. It is necessary to pay attention to the effect of aircraft bending and vibration on the distributed system and to provide a rigid mounting structure.

Another problem to be solved is the making of safety critical and mission critical channels independent, because a failure in the mission critical part must not influence the safety critical outputs. This can be solved by sharing no components between the channels and by separating instrument power and data channels both electrically and spacially. Furthermore the system must be protected against single-point failures and the cross-channel communications must be minimised. The use of resistors or diodes prevents failure propagation. All these methods lead to an increasing weight, power consumption and size.

5. Conclusions

Described in this paper are two steps for the integration of flight control sensors, AHRS and INS. It has been shown that the use of the same sensors for flight control and AHRS functions, as for example in the AMSU-system, yield excellent results. The AMSU-system has already demonstrated its performance capability in the EAP inherently unstable aircraft. It has to be stated that these aircraft, as opposed to conventional aircraft, require inertial sensor data with an extremely high bandwidth and very small data latency for flight control purposes.

The integration of flight control sensors and INS in one system is basically feasible

---

[17] RM - Redundancy Management

and has already been successfully implemented in commercial aircraft. The development of an integrated system using dithered laser gyros for inherently unstable aircraft involves considerable risks even when state of the art techniques are employed. This is due to the fact that such systems require a compromise between flight control requirements and navigation accuracy. Furthermore, systems of the second step of integration lead to no substantial saving in weight or space, as demonstrated by IISA.
Taking development risks into account, the cost of an integrated system is higher than that for separate systems or systems of the first integration step.

6. Acknowledgement

The author wishes to thank Messrs. Dr.J.Mark and M. Halverson for the opportunity to discuss in depth the problems associated with integrated systems design.

7. References

R. Ebner J. Mark	Redundant Integrated Flight-Control/Navigation Inertial Sensor Complex, Journal of Guidance and Control, Vol.1, No. 2, March-April, 1978, pp. 143-149
J. Mark R. Ebner A. Brown	Design of RLG Inertial Systems for High Vibration, PLANS 1982 Symposium pp. 379-385
D. Sebring J. Perdzock Captain Young	Application of Multifunction Strapdown Inertial System, AGARD, LS-No. 133, 1984
B. Stieler	Konzept eines Sensorsystems für zukünftige Jagd- und Kampfflugzeuge, Z. Flugwiss. Weltraumforsch. 9 (1985), Heft 6
J. Jankowitz D. Krasnjanski A. Glista	Joint Development of the Multi-Function Integrated Inertial Sensor Assembly (MIISA), 43rd Symposium of the Guidance and Control Panel on Advances in Guidance and Control Systems and Technology, London 7.-10.10.1986

DISCUSSION

J.Nicol, UK

The reliability of the described system is based upon hardware redundancy. What steps have been taken to ensure comparable software reliability, particularly in the microcode of unit processors?

Author's Reply

The three main steps of code walkthrough (i.e., module test, hardware integration, and software integration) have been accomplished. A computer-aided simulation was conducted to test all instructions implemented in the microprocessors. The test results have been documented in a test matrix that compares software requirements with test steps. The testing and intensive reviews covered about 50 percent of the software development.

DEVELOPMENT AND TESTING OF A PREDICTIVE METHODOLOGY FOR OPTIMIZATION OF MAN-MACHINE INTERFACE IN FUTURE AVIONICS SYSTEMS

Roger E. Parks
Group Engineer
Advanced Human Factors System Design
Bell Helicopter Textron Inc.
P.O. Box 482
Fort Worth, Texas 76101

SUMMARY

While new technologies offer the avionics designer opportunities in terms of providing systems with expanded capabilities, those opportunities are accompanied by new kinds of challenges and constraints which warrant a revision of traditional design life cycle strategies. The trend of for increasing complexity and cost in emerging avionics systems, driven by requirements for increased functional capability, has created a need for a predictive analytical methodology which: (1) accurately forecasts system performance early in the design process, and (2) treats the human operator and the equipment as a fully integrated man-machine system. A methodology which meets these needs has been developed and validated at Bell Helicopter Textron. The process is being used to provide early, accurate avionics system characterization, thereby reducing design costs.

THE PROBLEM

The applications of advanced technologies in combat aircraft offer the avionics designer both opportunities and constraints from a man-machine interface perspective. Many of the functions traditionally performed by the human operator can be automated. With innate human capabilities remaining essentially constant, the designer has the opportunity to provide more total system capability, improve combat mission effectiveness and provide the operator a workplace which optimizes his mental and physical capacities. The modern avionics designer must, in fact, exercise this opportunity. The demands of current and future battlefields dictate expanded man-machine system capability and performance.

Advanced technology applications also impose new kinds of complex challenges and constraints. These challenges and constraints are created, in part, by the pace of technology development and, in part, by a new category of problems which the advanced technology opportunities cause. The challenges and constraints-- "the problem" --is that the fielding of an advanced avionics system is, by and large, a race against time. The advanced system must remain effective against a battlefield environment which is highly dynamic. As threat capabilities change, so must system capability. If the design-to-fielding process is out-paced by battlefield demands, sub-optimum systems are fielded. While this system capability versus threat demand game of "leap frog" is ultimately a fact of life despite the time cycle of the development process, the application of advanced technology has created a new category of design issues which threatens to lengthen our processes even further. The nature of advanced avionics systems forces a close, near-complete functional integration of man and machine. Successfully coping with this level of integration goes beyond the traditional human factors engineering concerns associated with avionics design. The man-machine interface issues have become vastly more complex. The designer is, in fact, now confronted with the problem of predicting both man and machine performance, as both elements operate as one total, highly complex system. Predicting system performance is the issue that confronts us; efficiency in the design process is the objective.

Predicting the performance of the "machine part" of the man-machine system is relatively mature. Accurate predictions of how equipment will perform are quantifiable and normally done with confidence. However, when the human operator is added to the system, predictive methods traditionally become less quantifiable and subjectivity reigns. The traditional answer to this dilemma is, too often, a series of inefficient and expensive part-task studies, simulations and flight tests which, when compared to the quantification of machine performance, appears to approach trial and error. Compounding the problem is the fact that these kinds of human performance predictions traditionally occur well into the design process when system configuration changes are more costly. To avoid this situation, it is necessary to validly predict, at an early point in the design process, that both elements of the man-machine system will perform adequately to meet system performance requirements.

A NEW APPROACH

Traditionally, the fidelity of performance predictions has maintained a linear relationship with the design life cycle process. Performance predictions become increasingly valid as the design matures because adequate data is not available earlier. Unfortunately, the cost of design changes also increases with design maturity. This problem, illustrated in Figure 1, becomes more acute as the complexity and cost of avionics systems increases.

If more valid system performance predictions, based on more adequate data, are made early in the design process, the relationship between design changes, the design life cycle, and cost can be reversed. This concept is illustrated in Figure 2.

It is important to note (Figures 1 and 2) that the relative emphasis placed on analysis and man-in-the-loop simulation has increased. It is essential to understand that the increased analytical emphasis is more than simply increased time spent on the process. The analysis referenced is a methodology which has been developed by Bell Helicopter to support the design of advanced avionics systems which depend on optimized man-machine integration. This analytical methodology has been validated on several advanced avionics development programs and provides quantitative performance data for the aggregate man-machine system.

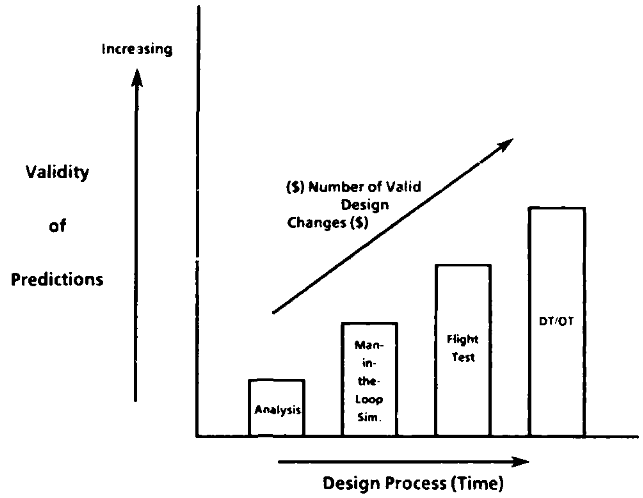


Figure 1. Traditional Design Approach

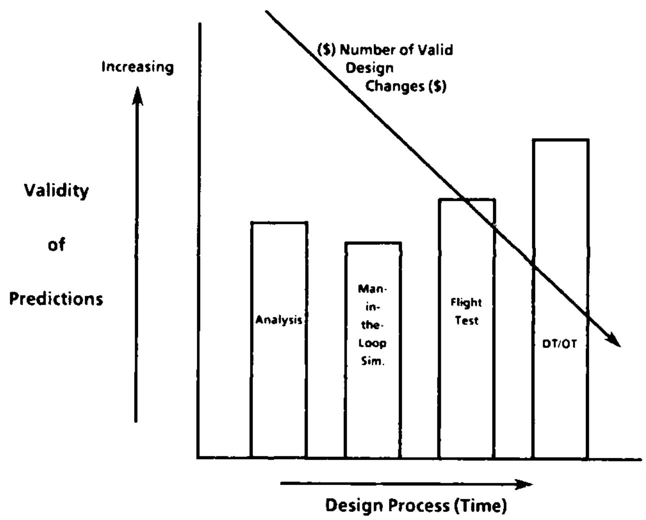


Figure 2. Suggested Design Approach

The methodology is unique in that it treats the crewmember(s) and the avionics equipment package as equal parts of a total system that must interact to perform a task. Human and equipment performance are quantified to determine: (1) if the man/machine system succeeded or failed in a given situation, and (2) why success or failure occurred. The five major elements of the methodology process and examples of predictive validity are discussed in the following narratives.

ELEMENT 1 - MISSION FUNCTION REQUIREMENTS ANALYSIS

The first element of the methodology is Mission Function Requirements Analysis. Analytical processes in this element include requirements from MIL-H-46855B combined with techniques from operations analysis, systems engineering and human factors engineering. Figure 3 provides a schematic representation of this element.

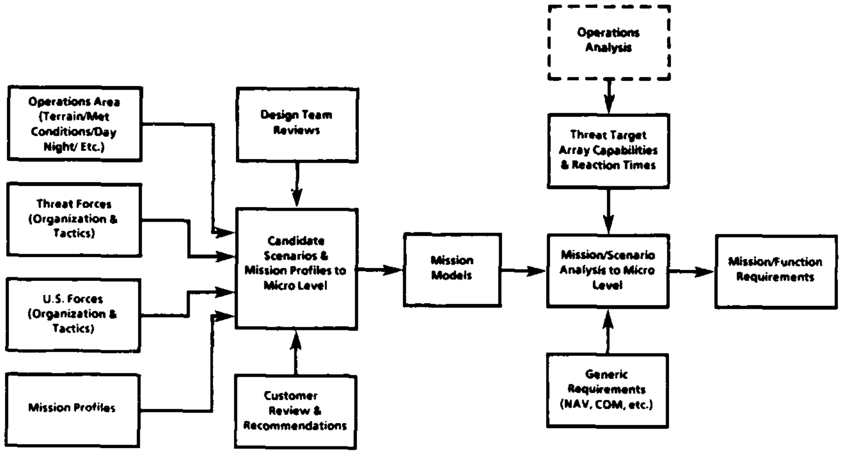


Figure 3. Mission Function Requirements

Detailed mission scenarios are developed that reflect the battlefield environment projected for the system being designed. As shown, candidate scenarios and mission profiles are based on a variety of inputs. The objective here is to develop an accurate real world arena in which the man-machine system will be evaluated. The accuracy of this characterization is critical and depends heavily on both subject matter expert involvement and a detailed understanding of threat capabilities and operational requirements.

Several mission profiles are normally needed to execute the total scenario. These profiles are constructed to reflect customer-defined missions, tactics, capabilities of the conceptual avionics equipment and airframe, and threat capabilities. Several detailed analytical battlefields are developed by locating threats in real world terrain and planning mission operations which reflect actual operational problems; e.g. weather, day/night, variant threat formations, etc. Profiles are sufficiently detailed to support analysis in one- to two-second time segments.

Mission profiles are digitally represented and analyzed via computer mission models. Models used contain terrain topography and feature data, threat formations, intervisibilities, etc.; and the mission profile. The results of this mission/scenario analysis (Figure 3) include data such as detailed threat response times, threat and own-system acquisition windows, and response times required for the man-machine system to accomplish mission tasks.

This process allows the designer considerable flexibility in that a variety of conceptual equipment options may be characterized and evaluated against several battlefield situations. Candidate equipment arrays can be evaluated to determine factors such as required input/output, processing time available, alternate modes of operation and total operating timelines for the man-machine system.

ELEMENT 2 - CANDIDATE SYSTEM DEVELOPMENT AND GROSS TASK ANALYSIS

The second element of the methodology, Candidate System Development and Gross Task Analysis, refines the general mission function data developed in element 1. As shown in Figure 4, the candidate equipment package options are evaluated and refined and candidate equipment items are selected to work toward the objective of a baseline mission equipment package.

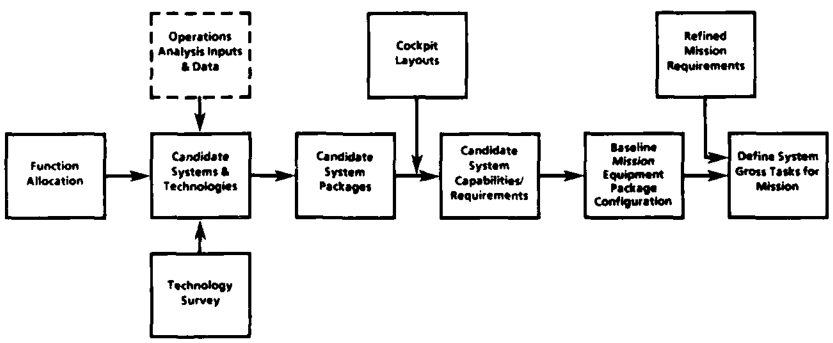


Figure 4. Candidate System Development and Gross Task Analysis

Early in the analysis, the candidate systems and technologies may range from real, i.e. state-of-the-art items, to purely conceptualized characterizations of out-year technology, depending on the design requirements at hand. From candidate systems and technologies, candidate system equipment packages are developed. Cockpit layouts and soft mockup studies add data to supplement the mission function requirements previously determined, and a baseline mission equipment package configuration is developed. To provide man-machine interface definition and prepare data for subsequnet man-machine system performance analysis, a gross task analysis is also completed at this point in the process.

A functional "road map" is created to provide a gross definition of both the flow and task descriptors of each mission profile. The content of these block flow diagrams (Figure 5) is referred to as Level 1 Analysis.

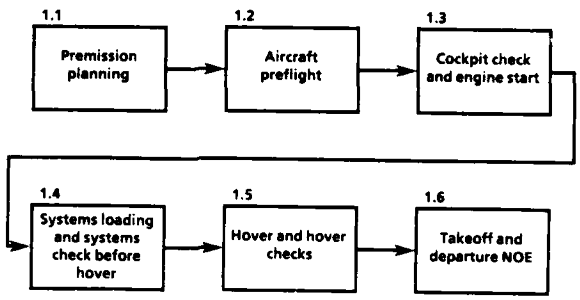


Figure 5. Example of Level 1 Analysis

Level 1 data is used to establish initial system design requirements for controls, displays and general boundaries for equipment operating timelines. When complete, the first element of the methodology establishes the man-machine system mission function requirements. Data available at this point in the process allows the designer to:

(1) Determine the top level functions the man-machine system must perform (e.g. communicate, navigate, etc.) and when those functions must occur in the profile.

(2) Define time boundaries that the man-machine system must function within to be mission effective.

(3) Characterize the equipment package at a gross level in terms of information input/output, processing requirements, etc.

During this phase of the analysis, gross tasks are identified and characterized to a second and third detail level. This process is illustrated in Figures 6 and 7. The Level 3 analysis characterizes tasks to a "check list" level and, as noted by referencing Figure 7, is near to becoming system specific.

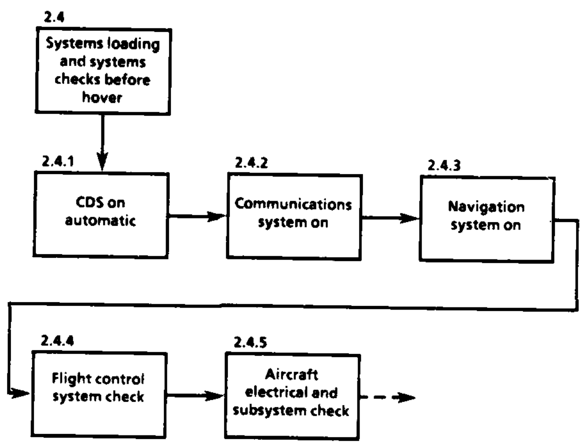


Figure 6. Example of Level 2 Analysis

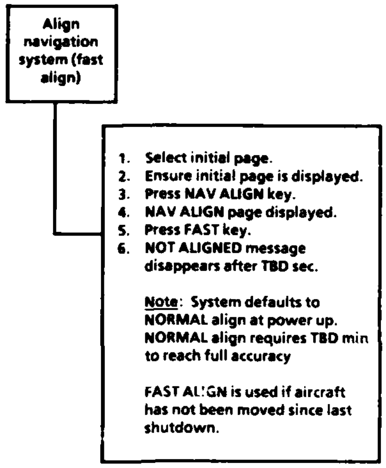


Figure 7. Example of Level 3 Analysis

ELEMENT 3 - CRITICAL TASK ANALYSIS

The third element of the methodology is critical task analysis. It is here that the analytical process definitely departs from the more traditional human factors processes in that the crew member(s) and the equipment package are treated as a single system. Both system elements (man and machine) have critical tasks that must be evaluated to accurately predict total system performance. The critical task analysis process is illustrated in Figure 8.

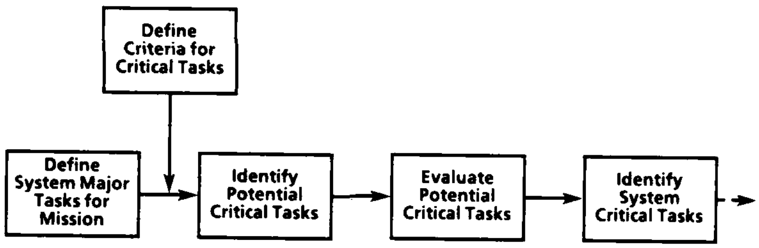


Figure 8. Critical Task Analysis

Critical tasks are selected on the basis of MIL-STD-46855B criteria; i.e. tasks which can adversely affect mission performance, system accuracy, or operator safety when either not performed or performed outside required time or sequence parameters. The rationale for concentrating on the critical tasks is that these tasks are the true system design drivers. The objectives of this analytical phase are to: (1) develop very detailed task descriptions, and (2) establish timelines for each task. To meet the first objective, each critical task is defined to its lowest operational element. An example of the lowest operational element or "subtask" for a crew member is a line address key selection on a multifunction display. A subtask example for an equipment item might be the field-of-regard scan time for a target detection system.

After critical tasks are defined to the required level, performance times and variances are determined for each subtask. This process is the basis for determining what must be performed by the man-machine system and how much time is required to perform it. At this point in the overall analytical process, critical subtasks have been identified and performance times for each have been determined. These data, coupled with the time-based forcing factors developed during the first element of the methodology, provide input to the man-machine performance analysis.

ELEMENT 4 - SYSTEM PERFORMANCE ANALYSIS

Element four of the analytical methodology employs a computerized system performance model to determine if the man-machine system can perform critical subtasks within required timelines. A schematic representation of element four is shown in Figure 9.

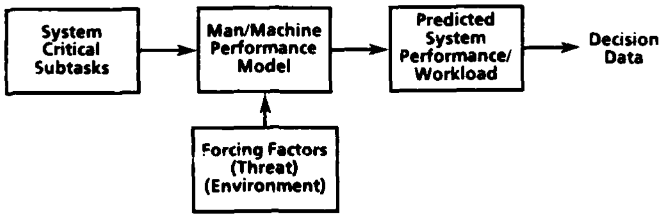


Figure 9. System Performance Analysis

The system performance model used is the Sequiter Workload Analysis System (SWAS).[1] This microcomputer based digital simulation model employs an expert system approach in its architecture, is relatively easy to use, and has the highly desirable features of minimizing subjectivity and allowing for analysis of concurrent tasks.

The SWAS model yields detailed outputs in both tabular and graphical format. Using this information, the analyst can make decisions regarding issues such as: (1) the probability of mission success, (2) the times required by both the crew member(s) and equipment suite to accomplish mission tasks, (3) the workload distribution between crew members (where multiple operators are involved), and (4) task "bottlenecks" generated by task time sharing overloads or delays caused by tasks not completed on time.

At this point in the analysis, the avionics system designer has quantifiable performance data which can be used to support a variety of decisions. For example, if the analytical data indicated that the man-machine system being considered could not complete all critical subtasks in the time available, the designer could examine the output data to determine where the time or task bottlenecks occurred. Since both human operator and equipment performance data are available, the designer could determine which part of the man-machine system was requiring excessive time and where, in the sequence of mission tasks, the time problem occurred. Since the performance model is computerbased, the designer is offered the flexibility of quickly re-running analyses of several alternative design strategies and selecting the optimum results.

Another feature of the analytical process is the ability to determine required crew size. The SWAS model provides for interactions of up to five crew members. The designer is, therefore, provided the opportunity to explore the crew size impact of task automation and ultimately determine the optimum man-machine mix.

ELEMENT 5 - VALIDATION

A major problem with analyses which accompany complex avionics systems development is having confidence in the analytical data. If a high level of confidence is achieved, simulation and flight test requirements can be reduced or, at worst, better planned. Bell's predictive methodology has been validated during several design programs via both man-in-the-loop (M-I-L) simulation and Flight test. As illustrated in the following figures, extremely accurate system performance predictions have been achieved.

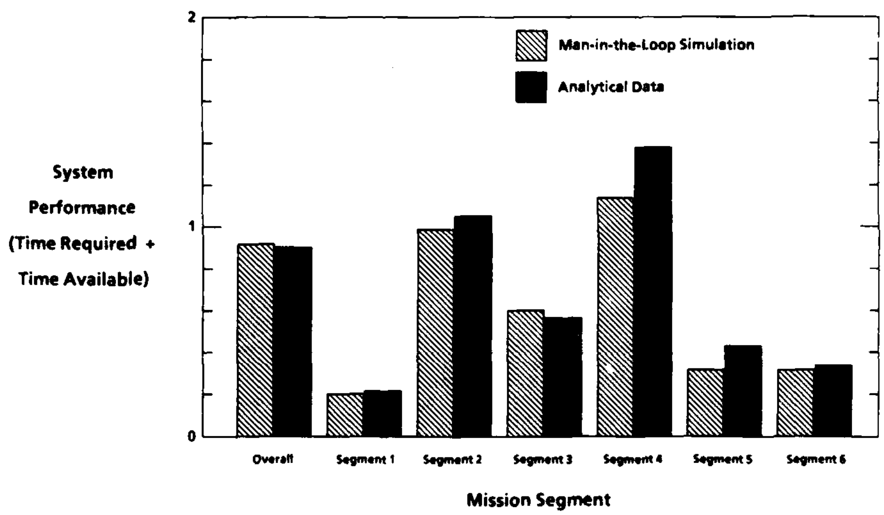


Figure 10. Predicted Versus Actual Performance

Figure 10 shows a comparison of predicted versus actual system performance data for six segments of a helicopter anti-armor mission. The system performance data (time required versus time available) was generated analytically, then collected during M-I-L simulation using six mission segments identical to the analytical segments. As shown, the predicted vs. actual results compare quite closely.

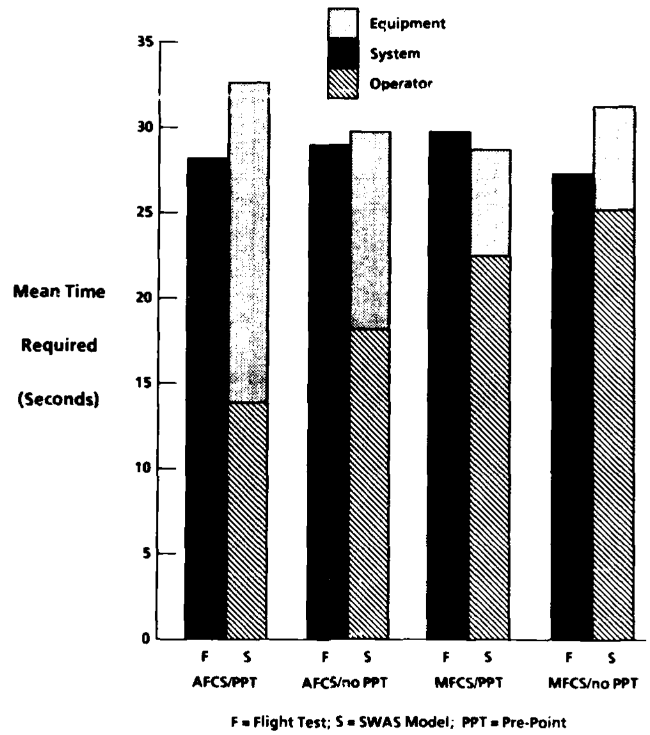


Figure 11. Predicted Versus Flight Test Data

Figure 11 shows a comparison of predicted performance data and actual flight test data for a series of tasks which evaluated pilot time performance during a nap-of-the-earth target acquisition exercise (with and without FLIR pre-pointing) while using automatic and manual flight control system modes. As in the previous example, the predicted and actual performance data compare very favorably. It is important to note that, in this example, the predicted time required data is identified as "equipment" or "operator" time. Recall that in the earlier analytical stages, critical task times are developed for both operator and equipment tasks. The delineation of times as shown is the manifestation of that process. This allows the designer to more accurately examine the significance of the man-machine division of labor for each task and assess the impact of alternate automation strategies. As a side note, the near-linear relationship between less system automation and increasing operator time required depicts what might be heuristically predicted; i.e., that operator workload increased as automation decreased.

Benefits to the Designer

Results obtained from the predictive methodology process provide the basis for several types of decisions that are integral to the advanced avionics design process. Examples of the data output and the kinds of decisions that data supports are shown in the following table.

DATA OUTPUTS	DECISION SUPPORTED
• Operator workload by mission segment	• Determination of required crew size • Workload demands by task
• Equipment workload by mission segment	• Does equipment selected/conceptualized meet mission demands? • What characteristics and capabilities must new equipment have?
• Detailed task time ratios (time required vs. time available)	• Is the man-machine system able to meet mission requirements?

While helping the designer make more valid and timely system configuration decisions reduces cost, more tangible efficiencies are realized by enabling a reduction in engineering simulation development and experimental flight testing. Prudent use of the predictive methodology offers opportunities to reduce the number of part-task developmental simulations and finalize full mission simulation configurations earlier. Part-task flight testing requirements can also be reduced. The need for manned soft-mockup studies can be eliminated in many cases. All such initiatives potentially improve design process efficiency and ultimately reduce the acquisition life cycle.

REFERENCES

1. SEQUITER'S WORKLOAD ANALYSIS SYSTEM; Sequiter Systems, Inc. Rt. 2, Box 907C5, Ft. Worth, TX 76135; 1986.

2. "A Mission Oriented Approach to Cockpit Design As Applied to Observation and Attack Helicopter"; R. R. Taylor, E. R. Poole; Bell Helicopter Textron; 1984.

ACKNOWLEDGEMENTS

The author wishes to acknowledge the support and sacrifices of his wife, Rebecca Ann, to whom he respectfully dedicates this paper.

DISCUSSION

G.Hunt, UK

In addition to data on the time taken for the aircrew to perform a task, simulation tests also produce data on the accuracy with which the result is achieved and on the difficulty experienced by the aircrew (e.g., the Cooper-Harper rating scale). Does the analytic model you described produce similar data?

Author's Reply

Yes, the methodology accounts for error rates, missed tasks, restarts, etc. Subjective data, such as Cooper-Harper, are used to supplement quantifiable results.

G.Bouche, GE

Did you develop a general guideline on how busy the operator should always be kept to achieve optimum performance in critical situations?

Author's Reply

No, a general guideline has not been developed on how busy the operator(s) should be kept. Optimum performance is very difficult to define, because it is based on individual differences and is highly situational.

B.H.Adams, UK

How do you account for the crew "scare level" in your analysis model?

Author's Reply

It is accounted for in terms of ranges of predicted performance. The "scare level" factor affects quantifiable performance in several ways that can be generalized and measured, although results are not exact.

J.Bart, US

Can this methodology be extended to account for design for maintainability (e.g., fault detection/isolation-repair) requirements and battle damage repair?

Author's Reply

Yes, in fact, this type of work is under way and has been validated.

R.J.Young, CA

(1) Determining crew size has, in the past, been quite subjective and often driven by a myriad of factors other than effective mission performance (e.g., size, weight, etc.). No quantitative data/models have been used extensively in the past to aid in establishing crew size.

(2) Intuitively, I think that single-crew aircraft in combat is a bad idea without "unacceptable" automation being necessary.

(3) Do you have any direct data now to indicate that a single-crew approach can be effective?

Author's Reply

Yes, although our results are limited to the rotary-wing community and combat helicopter programs, I would be glad to discuss them with you.

Also, I tend to agree with your second comment. "Acceptable automation" is the key phrase.

CREWSTATION INFORMATION AND DEVELOPMENT SYSTEM (CIDS)

by
M.E.Rowland and W.R.Wagoner
Boeing Military Airplane Company
P.O. Box 7730
Wichita, Kansas
67277-7730
U.S.A.

SUMMARY

This publication deals with the process by which requirements for an avionic system are translated into an integrated crewstation design. The Crewstation Information Development System (CIDS) has been divided into three phases of activity which can be summarized in the following paragraphs and illustrated in Figure 1.

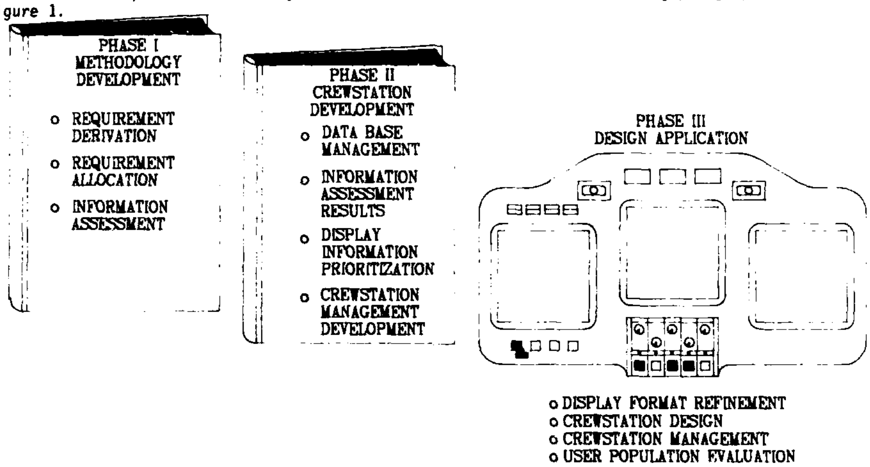


FIGURE 1 CREWSTATION INFORMATION DEVELOPMENT SYSTEM PHASE SUMMARY

Phase I, Methodology Development, develops the Crewstation Information Development Methodology (CIDS) which includes derivation of a comprehensive set of requirements, requirement allocation, and information utilization assessment.

Phase II, entitled Crewstation Development, focuses on deriving the most effective methods of utilizing required crewstation information taking into consideration the impact of the operational environment.

The final phase, Design Application, concerns the details of crewstation design, and the development of a crewstation information manager.

These three phases of activity represent a systems engineering approach which incorporates the disciplines of hardware engineering, software engineering and human engineering. The result is a crewstation design which addresses operator and mission requirements in a manner which enhances the man machine interface.

PREFACE

The complexity of projected battlefield operations requires increasingly sophisticated avionics systems and crewstation layouts. The associated demands on the operator have escalated to the point of taxing his ability to rapidly assimilate critical information. Improving operator and system effectiveness mandates the development of new Control and Display avionics and the upgrading of existing aircraft avionics in a manner which results in an integrated crewstation. Systems are required to handle the vast amount of information necessary, and still remain flexible enough to accommodate future growth. With today's technology, subsystems tend to be treated individually with the result that a unified integration of the crewstation suffers. The Crewstation Information and Development System (CIDS), addresses these problems and provides the traceability and flexibility to evaluate requirement changes on a system level.

The intent of the CIDS three phase approach is to develop an avionic system architecture and crewstation operation derived from and respondent to crewstation needs. Specifically, through the development of a software model, top level requirements are expanded, refined, and singularly allocated to a control or display function in the crewstation. These allocations impose requirements on hardware and software capabilities, which in turn dictate a top level tailored Controls and Display architecture.

The subsequent development of distinctive display formats and adaptive crewstation controls based upon a methodical interactive approach provides a mission and operator effective crewstation design.

METHODOLOGY DEVELOPMENT (PHASE I)

In the early stages of a typical avionics development program, requirements are provided by the user in the form of overall system performance allocations or desired subsystem capabilities. In either case the designer is presented with a multitude of subsystem design goals which may or may not result in an overall satisfactory system performance. In many circumstances the pilot becomes the de facto system integrator of the various subsystem operations with the result that mission effectiveness suffers or workload increases. Therefore, the question becomes "How are requirements integrated into the crewstation in a manner which increases mission effectiveness and enhances operator workload?" This problem actually contains two distinct parts: 1) How can the crewstation designer incorporate all the various subsystem requirements in an integrated manner; and 2) assuming item one can be accomplished, what is the best way to present information and provide system control?

Phase I activity describes the process by which questions one and two above are answered. The CIDS Methodology Development as shown in Figure 2 is a result of this process. Representative outputs of this methodology will be presented in Phase II after the Crewstation Information Development System has been discussed. The Methodology Development discussion will incorporate the following major topics: Operational Requirements; System Designer Inputs; Information Assessment; and Information Assessment Implementation.

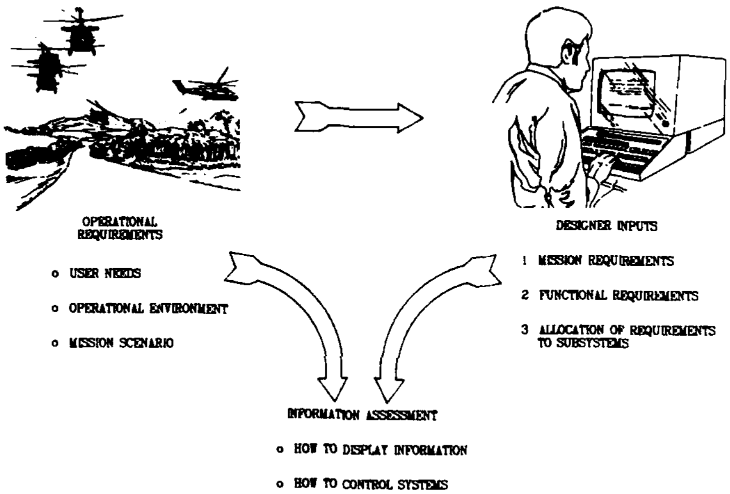


FIGURE 2 PHASE I CIDS METHODOLOGY DEVELOPMENT

OPERATIONAL REQUIREMENTS

With reference to Figure 2, requirements are provided in the form of overall system performance and desired capabilities for a given operational environment. These requirements are translated into system attributes with the appropriate traceability to source documents. This process is the means by which overall mission requirements (which reflect the projected threat environment and enemy capabilities) are allocated to system operation functions.

SYSTEM DESIGNER INPUTS

Once this set of operational requirements is compiled, the system designer is able to expand and modify it to account for functional requirements not directly stated by the user. This activity can be seen in Steps 1 and 2 of DESIGNER INPUTS in Figure 2. These requirements supply the details of system operation and provide the framework by which subsystems interact in the accomplishment of overall system objectives. The allocation of requirements to subsystems, Step 3, is the process by which all requirements, whether directly stated or derived, are assigned to functional areas of the system architecture. This activity is particularly important in that it performs several major functions. First of all, it identifies where requirements will be performed. This simplifies the task of identifying multiple uses of common information, and eliminates duplication of sources. Additionally, the impact of one subsystem's requirements on another are readily seen. Next, the allocation of requirements determines to a large extent, the interfaces required between subsystems. These interfaces begin to identify control and display functions needed in the crewstation. Finally, this activity supports the definition of overall system operation, or what the roles of mission and system management are in the accomplishment of mission objectives and system operation.

These first three steps as described above are the means by which operator requirements drive the avionics architecture at the initial stages of system development. Allocation of requirements to specific

subsystems identifies needed hardware and software capabilities. These capabilities, will be reflected in the total avionic system architecture, the associated hardware elements, and the software processing. The ultimate target is a system initially aimed at and ultimately tailored to improved man-machine operation.

The activities accomplished to this point provide the ground work for the major thrust of the Crewstation Information Development System. The next task is the assessment of crewstation information for the purpose of determining how best to control or display it. This process is accomplished by critically evaluating each control and or display requirement and suggesting a design implementation. Steps 1-3 of Figure 2 facilitate the identification of those requirements which either must be controlled by the operator, or displayed to him. However, these same steps that allowed the assessment of crewstation information, also have had a direct impact on the overall system architecture.

INFORMATION ASSESSMENT

Information assessment takes the top level, derived, and lower level requirements and prioritizes them to arrive at an optimum implementation in the crewstation. This is accomplished by performing three tasks which are: the determination of when during the mission specific information must be controlled or displayed; the assessment of the critical nature of the information as it relates to overall system operation; and the assessment of the criticality of the information itself as it relates to the accomplishment of goals during specific phases of the mission. Once these tasks are accomplished for all the requirements, of all the subsystems, it is possible to assign specific control devices and display types to the crewstation. This process can be seen in Figure 3, Information Assessment Implementation.

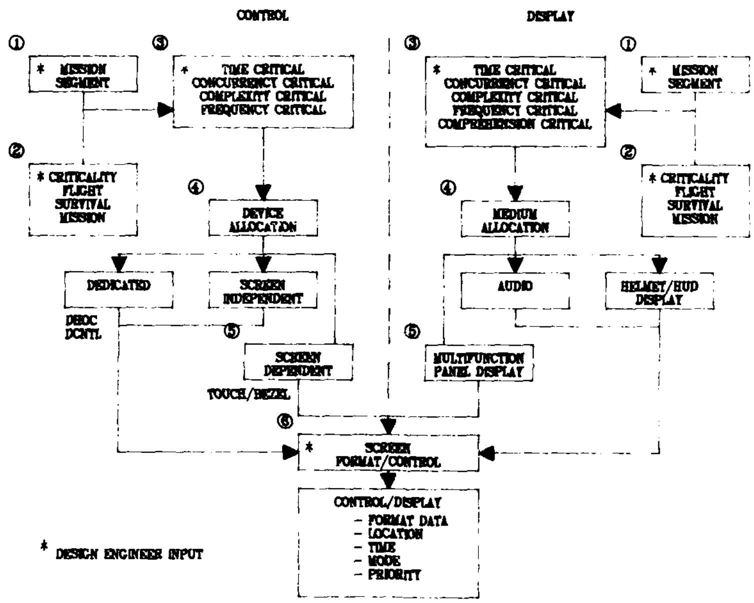


FIGURE 3 INFORMATION ASSESSMENT IMPLEMENTATION

INFORMATION ASSESSMENT IMPLEMENTATION

Steps 1 through 5 of the Information Assessment Implementation deal with functions requiring crewstation controls on the left of the figure and crewstation displays on the right of the figure. The circled numbers indicate the sequence of activity, and the asterisks specify those areas where inputs from the crewstation designer are required. As stated earlier, the first step of the crewstation requirement implementation is the determination of when during the mission segment previously identified control and or display functions are needed. For the purposes of this paper we shall consider a combat mission with the following mission segments:

- o Pre-take-off
- o Take-off
- o Enroute
- o Air-to-air/air-to-ground
- o Egress
- o Landing
- o Shut down

In addition to determining when to display or control information, the criticality of that information must also be assessed. Assessing the criticality of crewstation information is a two part task which corresponds to Steps 2 and 3 of the Information Assessment Implementation (Figure 3). Step 2 involves determining if the function is flight, survival, mission, or mission enhancement critical. The definition of these criticalities is stated below on the next page and roughly equates to that used in MIL-STD-882, System Safety Program Requirements.

DISPLAY TYPES	CONTROL DEVICES
o Alphanumerics (A)	o Dedicated hands on control (DHC)
o Graphics (G)	o Programmable display push button (PDP)
o Symbology (S)	o Dedicated Control (DCL)
o Artificial voice response (V)	o Voice (VRR)
	o Touch screen/bezel (FOG)

Each of these ways of displaying information to the crewmember or controlling system functions has its unique numerical value when evaluated according to the critical parameters. A summary of this evaluation can be seen in Table 1. By totaling the individual ratings of symbology, voice reponse, graphics, and alphanumerics, it is possible to numerically rank display types. This numerical ranking suggests that certain ways of displaying certain types of information are more appropriate than others. For example, symbology ranks high (numerical value 5) for time criticality (TCRIT) and comprehension (CMPH). However, it is not as suitable where complexity is concerned (CMPX value of 4). Alphanumeric on the other hand has a high value for complexity, but low values for time criticality and comprehension. Using this criteria it is possible to compare the numerical ranking of the crewstation requirements to that of display media in order to suggest the most appropriate control or display means. The problem with this direct comparison is that it does not account for the importance of flight, survival, mission, and mission enhancement criticalities during specific mission segments. To account for these influences on control and display implementation, weight factors which correspond to the mission segment and flight criticality of the requirement are factored into the numerical value of its assigned critical parameters.

TABLE 1 DISPLAY MEDIA/CONTROL DEVICE CRITICALITY

DISPLAY MEDIA	TCRIT	CON	CMPX	FREQ	CMPH	TOTAL
SYMBOLOGY (S)	5	4	4	4	5	22
VOICE (RESPONSE (V)	4	5	2	5	3	19
GRAPHICS (G)	3	2	3	3	4	15
ALPHANUMERICS (A)	2	3	5	2	2	14

CONTROL DEVICE	TCRIT	CON	CMPX	FREQ	TOTAL
DEDICATED HANDS ON CONTROL (DHC)	5	5	1	5	16
PROGRAMMABLE DISPLAY PUSHBUTTON (PDP)	3	1	4	3	11
DEDICATED CONTROL (DCL)	2	4	2	1	9
VOICE (VRR)	4	3	3	4	14
TOUCH SCREEN (FOG)	1	2	5	2	10

These weight factors are derived from the following three technical papers: "Artifical Intelligence Approaches in Intelligent Helicopter Automation for Nap-of-the-Earth Missions;" "Flight Management for Air-to-Surface Weapon Delivery;" and "Automation in Combat Aircraft." In all of these papers experts were asked to evaluate specific mission segments and functions that have the largest impact on mission objectives. The summation of the results of these papers is illustrated in Table 2. Interestingly enough, the importance of specific mission segments in the accomplishment of flight, survival and mission objectives does not appreciably vary from fixed wing to rotory wing aircraft missions.

TABLE 2 WEIGHT FACTORS

MISSION SEGMENT	CRITICALITY			
	FLT	SUR	MSN	MSNE
PRE-TAKE OFF	.2	.6	.4	.4
TAKE OFF	.5	.4	.4	.4
ENROUTE	.7	.7	.5	.5
AIR-TO-AIR/AIR-TO-GROUND	.9	.8	.7	.7
EGRESS	.7	.8	.5	.5
LANDING	.5	.4	.4	.4
SHUTDOWN	.2	.2	.1	.1
AVERAGE	.58	.56	.43	.43

The final aspect of Phase I Methodology Development concerns the method by which crewstation requirements are allocated to control devices or display information types (symbology, alphanumerics, graphics etc.). Essentially, what this entails is matching the numerical total of a critically assessed requirement biased by an appropriate weight factor (Table 2) against a prioritized list of control devices or display information types (Table 1).

For example, assume that a display requirement has a total of 17 when evaluated according to the five critical parameters discussed earlier (TCRIT, CON, CMPX, FREQ, and CMPH). Also, assume that the display requirement is survival critical (SUR) and is displayed during an enroute mission segment. The weight factor for this display requirement is 0.7 (Table 2). If we multiply 17 times 0.7, the product is 11.9 which does not immediately suggest a display type when compared to the TOTALs of Table 1. To resolve this dilemma, a range of values which would provide a suggested control device or display type when compared to a requirement's weighted critical assessment was developed. The results of this development can be seen in Table 3, Display Type Allocation Values.

- Flight Critical (FLT) - a control action or display function which if lost or degraded would seriously impair or negate the pilot's ability to fly the aircraft.

- Survival Critical (SUR) - a control action or display function which directly or indirectly affects the ability of the aircraft to survive in a hostile environment.

- Mission Critical (MSN) - a control action or display function which affects the performance of mission objectives.

- Mission Enhancement (MSNE) - control actions or display functions which if lost would not automatically nullify the accomplishment of mission objectives, but may necessitate degraded mode operation or work around implementation.

The final task of criticality assessment involves the evaluation of crewstation requirements against five parameters for display functions, four of which are also used for control functions. These parameters are used to determine the criticality of a control action or display function as it relates to the accomplishment of mission segment tasks previously allocated. These parameters are defined below, the first four of which pertain to both control and display assessment, and the final one which deals exclusively with display assessment. A rating scale follows each critical parameter, to provide a means of quantitatively rating each crewstation requirement.

- Time Criticality (TCRIT) - the requirement for control actions or display information to be immediately available during a mission segment.

  1 - Improbable

  2 - Remote

  3 - Occasional

  4 - Probable

  5 - Frequent

- Concurrency (CON) - the requirement for control functions or display information to be available concurrently due to mission segment demands. The higher the incidence of concurrency the greater the need for immediate access.

  1 - Improbable

  2 - Remote

  3 - Occasional

  4 - Probable

  5 - Frequent

- Complexity (CMPX) - the degree of difficulty in performing a control function or in assimilating display information.

  1 - Elementary

  2 - Simple

  3 - Difficult

  4 - Complex

  5 - Very complex

- Frequency (FREQ) - the number of times during a particular mission segment that control actions or display information will be required.

  1 - Improbable

  2 - Remote

  3 - Occasional

  4 - Probable

  5 - Frequently

- Comprehension (CMPH) - the difficulty of specific display information to be understood, which is to a large degree dependent on the required accuracy of the information. For example, bank angle may not require any great degree of precision whereas final approach airspeed does.

  1 - Trivial

  2 - Simple

  3 - Moderate comprehension difficulty

  4 - Difficult to comprehend

  5 - Most difficult to comprehend

Once all of the crewstation requirements have been assigned to mission segments and prioritized according the preceeding definitions, the control device and display type can be selected. These first three steps can be accomplished by the crewstation designer or by the user. If the latter approach is used several benefits will occur. First of all, the inclusion of the user at this early stage of development will provide authenticity as well as valuable operational inputs which may not be readily available to the crewstation designer. Also by allowing the operational user the opportunity to evaluate the importance of control actions and display information, it is possible to tailor the crewstation to various user preferences.

With the initial inputs for the Crewstation Information Development System completed, the process by which control devices and display types are selected can be discussed. This process involves the evaluation of several display types according to the same critical parameters (TCRIT, CON, CMPX, FREQ, and CMPH) discussed previously. The rating of these display types can then be matched against the individual crewstation requirements to determine a design implementation. The display types and control devices are:

The derivation of the limits for each criticality can be described in four steps: 1) upper limit; 2) lower limit; 3) upper intermediate limit; and 4) lower intermediate limit. We will use flight criticality of display types as an example.

> Step 1 deals with the upper limit of flight criticality, 22.50. Since there are five possible information assessment categories (time, concurrency, complexity, frequency, and comprehension), and the highest value for each is 5, the highest total of these is 25. The highest weight factor for flight critical is 0.9 (Table 2). Therefore, the highest value that a flight critical display requirement could be is 25 times 0.9, or 22.5.
>
> The derivation of the lower limit of flight criticality, Step 2, follows the same reasoning as Step 1. The lowest possible total value for the information assessment is 5. The lowest possible flight critical weight factor is 0.2. Therefore, the lower limit is 1.00.
>
> Step 3 deals with the derivation of intermediate limits. Symbology has the largest total value of the display types listed in Table 1. If we multiply this number, 22, times the average weight factor for flight critical, 0.58, we obtain 12.76.
>
> The required range for flight critical symbology therefore is 12.76 to 22.5. The upper limit for the next highest display type, Step 4, (voice, total = 19, Table 1) is 0.01 below the lower range for symbology or 12.75. Taking the average weight factor for flight critical (0.58), times the total for voice (19) gives the lower range for voice, 11.02. Therefore, the range for voice is 11.02-12.75.

TABLE 3 DISPLAY TYPE ALLOCATION VALUES

DISPLAY MEDIUM	FLIGHT CRITICAL (FLT)	SURVIVAL CRITICAL (SUR)	MISSION CRITICAL (MSN) MISSION ENHANCEMENT (MSNE)
SYMBOLOGY (S)	22.50 – 12.76	20.00 – 12.32	17.50 – 9.46
VOICE (V)	12.75 – 11.02	12.31 – 10.64	9.45 – 8.17
GRAPHICS (G)	11.01 – 8.70	10.63 – 8.40	8.16 – 6.45
ALPHA (A)	8.69 – 8.12	8.39 – 7.84	6.44 – 6.02
OPTION (OPT)	8.11 – 1.00	7.83 – 1.00	6.01 – .05

TABLE 4 CONTROL DEVICE ALLOCATION VALUES

CONTROL DEVICE	FLIGHT CRITICAL (FLT)	SURVIVAL CRITICAL (SUR)	MISSION CRITICAL (MSN) MISSION ENHANCEMENT (MSNE)
DEDICATED HANDS–ON CONTROL (DHC)	18.00 – 9.26	16.00 – 8.91	14.00 – 6.86
VOICE (VRR)	9.25 – 8.11	8.90 – 7.80	6.85 – 6.01
PROGRAMMABLE DISPLAY PUSHBUTTON (PDP)	8.10 – 6.37	7.79 – 6.13	6.00 – 4.72
TOUCH SCREEN (FOG)	6.36 – 5.79	6.12 – 5.57	4.71 – 4.29
DEDICATED CONTROL (DCL)	5.78 – .80	5.56 – .80	4.28 .40

Referring back to the example described earlier, a survival critical display requirement with a total of 17, a weight factor of 0.7, and a product of 11.9, would result in the selection of voice as the optimum display type (see Table 3). Table 4 provides the control device allocation values. The derivation of these values is identical to that of display types.

The development of the CIDS Methodology which facilitates the assessment of crewstation requirements, completes the activity of Phase I. Phase II, Crewstation Development discusses the results of methodology implementation and the beginnings of a crewstation information management function.

CREWSTATION DEVELOPMENT (PHASE II)

The second phase of the Crewstation Information and Development System deals with: data base management; information assessment results; display information prioritization; and crewstation management development. The major emphasis of the second phase, is the discussion of the results obtained through the implementation of the CIDS Methodology developed in Phase I. The discussion of these results will be conducted in six parts. These parts include: source requirements; functional requirements; requirement allocations; detailed allocation; information assessment values; and information allocation. Methods of prioritizing information in the crewstation will be discussed with the goal of providing a flexible method of dealing with a dynamic operational environment. Finally, using the results of information assessment, we will briefly discuss the initial development of a crewstation information manager. However, before the methodology results are presented a brief comment concerning data base management is appropriate.

DATA BASE MANAGEMENT

When CIDS was initially conceived, the size, complexity and intricacies of the data base were not imagined. What began as an effort to develop an integrated crewstation design with an equivalent data base, became an entire avionic system with its associated data bases. This resulted from the desire to develop a realistic design effort which could, at some point, become a generic model for a system engineer tool. The data bases expanded to include total system requirements, as as this happened the need to develop a common data base which included all the significant activities of the CIDS Methodology became apparent. The software used throughout Phase I and II was PC FOCUS which is a comprehensive information control system with some arithmetic applications and a limited graphics capability. In

Phase I of CIDS, data was processed for each step of the development methodology. This was accomplished by creating programs which retrieved required data from a preceeding step of methodology development, and inputting new data as required. This process became somewhat limited, in that data generated in one step could not automatically be used in a preceeding step if the need arose. This process also consumed a lot of overhead (duplicated data files and programs) and added to an already large data base. In short the need for a common data base has proven to be an important requirement which in future applications will be incorporated.

INFORMATION ASSESSMENT RESULTS

In order to present the results of CIDS in a logical manner, we shall follow the sequence of activity used in the discussion of the CIDS Methodology. Representative data will be presented in the form of Tables, and for the purpose of clarity we will deal with one subsystem of an avionic architecture, Armament Control.

SOURCE/FUNCTIONAL REQUIREMENTS

Table 5, Source/Functional Requirement, shows requirements which as a starting point have initially been allocated to the Armament Control subsystem. At this point, the assignment of a requirement to a subsystem is somewhat arbitrary but does provide an organizational tool for the disposition of all user stated capabilities. The first column of Table 5 shows the requirement numbers assigned to each requirement. These numbers are the means by which requirements are traced to source documents. In addition, requirements which are not directly stated by the user, and those requirements which are needed to support subsystem functions, are also documented to provide traceability.

TABLE 5 SOURCE/FUNCTIONAL REQUIREMENTS
ARMAMENT CONTROL

REQ NO	REQUIREMENT	DERIVED REQMT	STATED REQMT	FUNCTIONAL REQMT
300.00	MAN/AUTO CUEING/TRACKING OF VEHICLES AND AIRCRAFT	DERIVED		
301.00	REQUIRED FUNCTIONS			
301.02	RANGE FINDING	DERIVED		
301.03	DESIGNATION	DERIVED		
301.04	LAUNCHING (MISSILES, ROCKETS)	DERIVED		
301.05	FIRING (GUN)	DERIVED		
302.00	STORES JETTISON	DERIVED	2.3.2.1.3.6	
304.00	BALLISTIC CALCULATIONS	DERIVED		
304.01	DETERMINE CCIP			FUNCTIONAL
304.02	DETERMINE CCRP			FUNCTIONAL

Requirement numbers, in groups of 100, are allocated to each subsystem so that the assignment of numbers to requirements provides an initial subsystem allocation. For example, Armament Control is assigned the number block from 300.00 to 399.00. Therefore, a requirement stated by the user or derived from the designer, which initially appears to fit in the Armament Control subsystem would be given a number from 300.00 to 399.00. The decimal places to the right of the decimal indicate top level and lower level requirements. Two zeros indicate a top level requirement, and XXX.01 thru XXX.99 indicate lower level requirements related to the top requirement. It is important to note that this system provides the capability to assess the impact of requirement changes, determine degraded mode operation, and indicate how requirements have been satisfied in the design. These capabilities are possible due to the fact that, throughout the CIDS Methodology, a requirement retains its originally assigned number with more decimal places added to the right as the requirement is expanded.

REQUIREMENT ALLOCATION

Once requirements have been assigned a number and initially allocated to subsystems, they are evaluated as to how they interact with all the other subsystems. Initially this consists of determining where the performance of the requirement in question is accomplished. Referring to Table 6, a "P" under the appropriate subsystem column indicates that this subsystem is the source of the required information. An "O" followed by a letter corresponding to a subsystem, indicates an output from that column to the subsystem indicated by the letter. For example, requirement number 301.04 shows an "OW" in the CPT (cockpit) column. This means that there is an output from the cockpit to the armament subsystem (W). In addition, the accomplishment of this requirement is performed in the armament subsystem, indicated by a "P" in the "ARM" column. The last column of the table (C&D) shows that requirement 301.04 has both a control function and a display requirement indicated by "C/D". The identification of these control functions and display requirements provides the data required for information assessment and information allocation steps of the CIDS process.

TABLE 6 REQUIREMENT ALLOCATION AND INTERFACE
ARMAMENT CONTROL

REQ NO	REQUIREMENT	MSN MGT (M)	SYS MGT (S)	COMM (C)	NAV (N)	FLT CTL (F)	ARM (W)	ASE (A)	TGT ACQ (T)	CPT (H)	NVP (V)	PROC (K)	C&D
300.00	MAN/AUTO CUEING/TRACKING OF VEHICLES AND AIRCRAFT						P						C/D
301.00	REQUIRED FUNCTIONS						P						
301.04	LAUNCHING (MISSILES, ROCKETS)						P			OW			C/D
301.05	FIRING (GUN)						P			OW			C
302.00	STORES JETTISON						P			OW			C/D

Once all the requirements have been allocated to a subsystem, and an initial cut at the subsystem interfaces has occurred, a sort of all the inputs, outputs, and assigned requirements for a subsystem is accomplished. The assembled data from this computer sort contains all the information concerning the functions performed by that subsystem, and the information required from other subsystems to support them. This data provides the system designer the ability to verity proper subsystem to subsystem communication. In addition, support data required by one subsystem from another is immediately indicated in the sorted report data of both affected subsystems. This is important due to the fact that subsystem design is usually allocated to separate design groups, and as such the needs of one group are not always efficiently communicated to another.

DETAILED ALLOCATION

The allocation process, which was initiated with the allocation of requirements to subsystems, continues in this next area with the allocation of requirements to software modules and hardware line replaceable units (LRUs). In addition, unit values, frequency requirements, accuracies, and range data are developed. This information is illustrated in Table 7, and provides additional requirement detail to assure acceptable performance. Through this activity and that of the preceeding section, the crewstation requirements make their impact on the other subsystems and on the total avionic architecture.

TABLE 7 DETAILED INFORMATION ALLOCATION
ARMAMENT CONTROL

REQ NO	VARIABLE	SOFTWARE MODULE	HARDWARE LRU	UNITS	ACCURACY	RATE
300.00	MAN/AUTO CUEING/TRACKING OF VEHICLES & AIRCRAFT	FCCP		DEG:MIN:SEC	TBD	30 HZ
301.00	REQUIRED FUNCTIONS	FCCP				
301.01	RECOGNITION	FCSCP		DIGITS		16 HZ
301.02	RANGE FINDING	FCSCP	LRF	METERS	TBD	16 HZ
301.03	DESIGNATION	FCSCP	LD	DIGITS		16 HZ
301.04	LAUNCHING (MISSILES, ROCKETS)	FCSCP	AIU	DIGITS		16 HZ

INFORMATION ASSESSMENT VALUES

With the assignment of all subsystem requirements to either a "C", "D" or both "C&D" it is possible to assess control and display functions separately for the purpose of assigning criticalities. The assignment of these values can be seen in Table 8, Display Information Assessment Values, and Table 9, Control Assessment Values.

TABLE 8 DISPLAY INFORMATION ASSESSMENT VALUES
ARMAMENT CONTROL

REQ NO	REQUIREMENT	MISSION SEGMENT	FLT	SUR	MSN	MNSE	TCRIT	CON	CMPX	FREQ	CMPH	WHERE	WHEN	DCRIT	DCON	DCMPX	DFREQ	DCMPH	C&D
300.00	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	ENROUTE		X			5	5	4	4	4	1/2	CON	S	V	S	S	G	C/D
300.01	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	A-A/A-G		X			5	5	4	5	4	1/2	CON	S	V	S	V	G	C/D
300.03	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	EGRESS		X			5	5	4	4	4	1/2	CON	S	V	S	S	G	C/D
301.010	RECOGNITION	ENROUTE		X			4	4	5	4	5	1/2	M-C	V	S	A	S	S	D
301.011	RECOGNITION	A-A/A-G		X			5	5	5	5	5	1/2	CON	S	V	A	V	S	D

TABLE 9 CONTROLS ASSESSMENT VALUES
ARMAMENT CONTROL

REQ NO	REQUIREMENT	MISSION SEGMENT	FLT	SUR	MSN	MNSE	TCRIT	CMPX	CON	FREQ	TCRIT CNTL	CON CNTL	CMPX CNTL	FREQ CNTL	C&D
300.00	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	ENROUTE		X			4	3	3	2	VRR	VRR	VRR	FOG	C/D
300.01	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	A-A/A-G		X			4	3	4	4	VRR	DCL	VRR	VRR	C/D
300.03	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	EGRESS		X			4	3	3	2	VRR	VRR	VRR	FOG	C/D
301.020	RANGE FINDING	ENROUTE		X			4	3	4	2	VRR	DCL	VRR	FOG	C/D
301.021	RANGE FINDING	A-A/A-G		X			4	3	5	5	VRR	DHC	VRR	DHC	C/D

Table 8 shows the assigned requirement number, the requirement itself, the applicable mission segment, the flight criticalities (FLT, SUR, MSN, and MSNE as defined in the paragraph "Information Assessment Implementation"), the information criticalities (TCRIT, CON, CMPX, FREQ, and CMPH), where the information is to be display (where), when it is to be displayed, the display type assignment as a result of the information criticality (DCRIT, DCON, DCMPX, DFREQ, and DCMPH), and the control and/or display determination (C&D). For example, requirement number 301.11 is required during the air-to-air (A-A) or air-to-ground (A-G) engagement mission segment and is survival critical. The assignment of "5" to all the information criticalities corresponds to the selection of different display types (S - symbology, V - voice, A - alphanumeric, V - voice, and S - symbology). These selections agree with the data presented in Table 1 and represent display suggestions based upon a criticality without regard to other moderating factors. The WHERE column indicates a suggested display location (1 = head-up,

yes-up, 2 = head-up, eyes-down, 3 = head down, and 1/2 either 1 or 2) and is based upon the flight criticality of the display requirement. The WHEN column provides information about when to display information to the crewmember. The choices available are: CON continuously; REQ - on request; M-C - mode dependent continuously; and M-R - mode dependent on request. The derivation of these assignments is based on the numerical total of time criticality (TCRIT), frequency (FREQ), and concurrency (CON). The range for constant (CON) is 14-16, for mode constant (M-C) is 9-13, and for mode on request in 1-8.

Table 9, Control Assessment Values, provides similar information as that of Table 8 with the exception that "when" and "where" are not directly addressed but are a function of the control device used. It is significant to note that the options supplied for control devices and display types are not sufficiently reduced to allow a definitive choice. However, using the information developed in Tables 3 and 4 the optimum choice for both display type and control device can be made. The results of employing Tables 3 and 4 to suggest a control device or display type can be seen in Tables 10 and 11. The display type and control device suggestion are accomplished by comparing the product obtained from the total of the information criticalities and a weight factor with the values of Tables 3 and 4. The result of this process is shown under the "CHOICE" column of Table 10, and the "1ST PIK" column of Table 11. Additionally provided is a "2ND PIK" for control device to provide an alternate means of crewstation control. Both Tables show the information critical parameter which had the greatest impact on the display type or control device selection. The significance of the data in Tables 10 and 11 is that display formats can be developed for specific areas (head-up eyes-out, head-up eyes-down) of the crewstation, and for specific times during the mission (take-off, enroute, etc.).

TABLE 10 DISPLAY MEDIA ALLOCATION
ARMAMENT CONTROL

REQ NO	REQUIREMENT	MISSION SEGMENT	FLT	SUR	MSN	MNSE	WHERE	WHEN	WT FAC	PROD	TICRT	CON CRT	CON PARA CRIT	FREQ CRIT	CMPH CRIT	CHOICE
300.00	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	ENROUTE		X			1/2	CON	.7	15.40	TI	CON				SYMBOL
300.01	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	A-A/A-G		X			1/2	CON	.8	18.40	TI	CON		FREQ		SYMBOL
300.03	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	EGRESS		X			1/2	CON	.8	17.60	TI	CON				SYMBOL
301.010	RECOGNITION	ENROUTE		X			1/2	M-C	.7	15.40			CMPX		CMPH	SYMBOL
301.011	RECOGNITION	A-A/A-G		X			1/2	CON	.8	20.00	TI	CON	CMPX	FREQ	CMPH	SYMBOL

TABLE 11 CONTROLS DEVICE ALLOCATION
ARMAMENT CONTROL

REQ NO	REQUIREMENT	MISSION SEGMENT	FLT	SUR	MSN	MSNE	WT FAC	TOTAL	PROD	TIME CRIT	CON CRIT	CMPX CRIT	FREQ CRIT	FREQ	1ST PIK	2ND PIK
300.00	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	ENROUTE		X			.7	12	8.40	TIME				2	VRR	PDP
300.001	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	A-A/A-G		X			.8	15	12.00				ECRT	4	DHC	VRR
300.002	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT			X			1.0	15	15.00				ECRT	4	DHC	VRR
300.003	MAN/AUTO CUEING/ TRACKING OF VEHICLES & AIRCRAFT	EGRESS		X			.8	12	9.60	TIME				2	DHC	VRR
301.020	RANGE FINDING	ENROUTE		X			.7	13	9.10	TIME	CON			2	DHC	VRR

One final aspect of Phase II activity is the prioritization of display information. The optimization of display information is necessary due to the large amount of data available to the crew. In order to assist the crew in displaying information relevant to the mission segment, a prioritization scheme was developed. This scheme involves assigning numerical values to the importance of a specific requirement as it relates to the accomplishment of mission segment tasks and the flight criticality. For example, high medium and low mission criticality (HCRIT, MCRIT, and LCRIT) would be assigned values of 3, 2, and 1. The definition of these mission criticalities is as follows:

- High Mission Criticality - loss of such information would seriously impair or prevent accomplishment of mission segment objectives.
- Medium Mission Criticality - information which impacts mission segment objectives but not to the extent of negating the accomplishment of these objectives, i.e. work arounds are necesary.
- Low Mission Criticality - information which does not have a direct bearing upon accomplishment of mission segment objectives, i.e. work arounds may suffice.

Also flight, safety and mission criticalities would be assigned a value of 3, 2, and 1 respectively. One value for each of the two sets of criticalities would be assigned to a requirement and the two numbers would be added for a maximum TOTAL of 6. As a result of this process, information can be sorted as a function of TOTAL for each requirement. Table 12 illustrates display information on a head-up, eyes-out display during air-to-air or air-to-ground mission segment for a TOTAL of greater than or equal to 5

during contour flight. The amount of information displayed has been significantly reduced, due to the fact that information can be selectively prioritized.

TABLE 12 DISPLAY INFORMATION AS A FUNCTION OF MISSION ACTIVITY ARMAMENT CONTROL

REQ NO	REQUIREMENT	MISSION SEGMENT	HALT	LALT	CON	NOE	AREA	HCRIT	MCRIT	LCRIT	FCRIT	SCRIT	MSN	TOTAL
302.002	STORES JETTISON	A-A/A-G	X	X	X		1	3			3			6
323.002	PROVIDE SELECTED STORES JETTISON COMMANDS	A-A/A-G	X	X	X	X	1		2		3			5
329.031	PROVIDE INDICATION OF HUNG STORE	A-A/A-G	X	X	X	X	1	3			3			6
329.051	PROVIDE INDICATION OF UNSAFE LAUNCH	A-A/A-G	X	X	X	X	1	3			3			6

The impact of this process is, with the user evaluating control and display requirements according to the critical parameters discussed in this paper, the crewstation can be tailored to individual needs. In addition the specific information that is shown, and how it is shown, can become a dynamic function of the operational environment.

With the discussion of display information prioritization completed, the description of Phase II activity is complete. Phase II presented the results of the CIDS Methodology Implementation in the form of representative outputs of the Armament Control subsystem. Phase III will discuss the next steps in the design implementation of the CIDS results.

DESIGN APPLICATION (PHASE III)

Phase III deals with the activity which has been initiated, but which at this point has not been completed. The major tasks of this phase include: display format refinement; crewstation design, which includes detailed display formats along with control device allocation; crewstation information management development; and design evaluation. The first three tasks are in various stages of development, and that development will be briefly discussed. The final task, design evaluation, will include a review of the criticality assessment values and the implementation of the CIDS Methodology into a crewstation mock-up.

DISPLAY FORMAT REFINEMENT

The process by which display information is allocated to specific areas of the crewstation was discussed in Phase II. The intent of this allocation was to generate a group of information display requirements for a specific crewstation location, and of specific display types (symbology, graphics, and alphanumerics), for the purpose of developing display formats. This process has been initiated with varying degrees of success. The major problem encountered was that the display information requirements were not sufficiently grouped to allow the formation of functional display formats. In other words, information which would typically be associated with system mode control and status, primary operation, or secondary operation did not readily fall out of the CIDS Methodology in order to facilitate functional grouping of display information. This functional allocation is in progress. Once all of the subsystem display information has been allocated to functional display formats, the formats will be developed.

CREWSTATION DESIGN

The second aspect of Phase III is the detailed design of the crewstation itself. The development of the display formats, along with a composite list of all the dedicated and non-dedicated crewstation controls, will enable the physical layout of the crewstation. In addition, by knowing what information is to displayed at what time and where, it is possible to determine the required number of displays and their location. This activity equates to the development of a physical crewstation which is the result of a methodical integrated approach.

CREWSTATION INFORMATION

The final area of Phase III activity, which is presently in work, is the development of a crewstation information manager. The intent of this effort is to develop a crewstation manager which takes the basic work done in CIDS and refines the prioritization of display information. To facilitate this effort a generalized "expert system" approach is being employed. Some of the goals of this activity are: demonstration of a prototype architecture concept; development of flexible formats for crewstation displays; development a scenario driver; development a graphics interface to a crewstation mockup; and demonstration of a working model within 12 months.

To accomplish these objectives a scenario was developed which provides a realistic, action packed description of a combat mission, which initially encompasses a 2 to 3 minute time frame. This scenario is intended to require "intelligence" either supplied by the crewmember of inherent in the software to provide task essential information. This "intelligence" necessitates the development of a variety of objects and rules. Work has begun to prepare a story board which will provide a description of the internal and external situation for time slices of the scenario. As a result of the story board activity, the data required for the scenario from the avionic systems will be identified and corresponding objects, attributes, and icons will be developed. Once this information is known (ie. objects which must be reasoned about), rules will be developed which take the data found in the scenario and create the required displays. These rules are derived from two sources, pilot expertise and display expertise. The pilot

expert provides instruction concerning what information is to be displayed (information priority), and when it is to be displayed. The display expert supplies how information is to be displayed, where it should be displayed, appropriate display size, display colors, use of voice, and any other pertinent data.

Using information about objects and displays from the scenario driver and the rulebase, graphic displays for a crewstation will be developed. Initially these displays will reside on one CRT attached to the work station, however, they will move into a crewstation mock-up as the system matures.

At the present time the development work station is in operation and the software development tool has been incorporated into it. Software training is complete and a baseline concept for display and airborne objects is complete. The specific scenario time slice has been identified and the development of the data structures is in progress. The first demonstration of this knowledge based information manager should be late June of 1987.

DESIGN EVALUATION

The process of evaluating a design generated from the Crewstation Information Management System is not something which is done once and then forgotten. It is an ongoing iterative activity which has been conducted several times already. Although the reviews conducted have been largely internal, the experience of employees with military aviation backgrounds has been employed. However, the major objective in the design evaluation is to encourage active military pilots to critically evaluate this system both in the assessment of control and display information criticalities, and in the crewstation design implementation. It is anticipated that a review of this nature will be accomplished within 18 months.

CONCLUSION

The discussion of the Crewstation Information Development System has been organized in three sections which correspond to the phases of system development.

Phase I presented the derivation of the CIDS Methodology. Phase II showed the results of the implementation of that methodology. Phase III discussed present activity and the direction of effort in the future.

The motivation for this project was the desire to provide a vehicle whereby the process of taking initial requirements and implementing them in the crewstation could be accomplished in a methodical way. In addition, the options available to the system designer in the implementation of requirements needed to be organized and prioritized so that an optimized design approach was suggested. From this process it is possible to develop a crewstation which encompasses display formats, display location, display content, display prioritization, when to display, and what controls to incorporate.

Throughout the three phases of activity of this effort some important lessons have been learned, the most important of which concerns the development of requirements. The requirements, as initially formatted by the designer, are used in every step of the CIDS Methodology. These requirements are allocated to several steps of the methodology where information concerning interfaces, display functions or control functions is required. This dictates that the requirement must be very specific, and have the capability to stand on its own. As such, the results of the implementation of the CIDS Methodology is highly dependent on the wording of the requirements used. Therefore, care must be taken in the initial development and formatting of requirements.

The application of this system has been directed at the development of a new avionic suite with a new crewstation. However, there are no apparent reasons why applications to existing aircraft requiring some modification, or ground vehicles, are not viable. Some modifications would be required to the existing data base, but the CIDS approach is workable. In order to facilitate a smooth transition to various other applications, the development of a generic requirements data base would be a valuable tool. This data base would include basic system functions such as navigation, communication etc, and the specific program application would add to, or delete from this generic source of requirements as required. In this regard considerable time and manpower savings could be realized in the development of an integrated system. This process can also provide a valuable tool to examine the effects of failures on crewstation operation and the level of redundancy required for critical crewstation functions.

In short, CIDS exhibits a great potential for supporting the development of comprehensive displays and information management for state-of-the-art weapon systems. The activity of Phase III will further identify and clarify the limits future application.

A CHANGE IN SYSTEM DESIGN EMPHASIS: FROM MACHINE TO MAN

M. L. Metersky
NAVAIRDEVCEN (Code 301)
Warminster, PA 18974-5000

J. L. Ryder
Pacer Systems, Inc.
4 Horsham Business Center
300 Welsh Road
Horsham, PA 19044
USA

SUMMARY

In the past, and to a large extent even today, the emphasis in system design has been on defining hardware requirements. In many cases, advances in hardware technology, and not the ability to meet mission requirements, were the driving factor that determined the need for upgrades to or replacement of a weapon system. Even though software has increased in importance and percentage of cost in system development, it is still philosophically considered as a means to facilitate hardware performance. This thinking has had an adverse effect on system performance by relegating decision requirements, which can be derived directly from mission requirements, to a minor or nonexistent role in the system design process. It is the premise of this paper that system design should be dictated by decision requirements since decisions humans make determine how well and to what degree a weapon system's inherent capabilities will be utilized. A system design approach based on an emphasis of the human in his role as a decision-maker is presented.

INTRODUCTION

Historically, even though the majority of systems designed and fielded have met their specifications, they often have not exhibited predicted performance. We believe this occurs because the role played by the human in the system, particularly in terms of decision making, has not been adequately considered in the design process. This inability to perform as expected is prevalent in weapon systems and exacerbated with command and control (C^2) systems. It is more pronounced as the level of decision dependency increases. Regardless of decision level dependency, current design philosophy ignores the human's role. We propose that it is necessary to reorient the system design approach, especially for C^2 systems, to stress the decision making function. As a consequence, system performance potential could be maximized by proper emphasis of the role played by humans.

In the past, and to a large extent even today, the emphasis in system design has been on defining hardware requirements. In many cases, advances in hardware technology, not the ability to meet mission requirements, were the driving factor that determined the need for upgrades to or replacement of a weapon system. Even though software has increased in importance and percentage of cost in system development, it is still philosophically considered as a means to facilitate hardware performance. This thinking has had an adverse effect on system performance by relegating decision requirements, which can be derived directly from mission requirements, to a minor or nonexistent role in system design. It is the premise of this paper that system design should be dictated by decision requirements since decisions humans make determine how well and to what degree a weapon system's inherent capabilities will be utilized.

In the decision-oriented approach proposed, the system is considered not as a package of hardware and software capabilities integrated to fulfill well-defined mission functions, but rather as a decision making system composed of three interacting elements--hardware, software and personnel. The design approach based on this viewpoint begins with mission requirements. From this, decisions necessary to perform specified mission functions, and hardware and software needed to support them, are then defined. This design approach focuses on maximizing the decision-making ability of an entire system by viewing the three system components as complementary and by integrating their capabilities synergistically.

RECENT TRENDS AND POTENTIAL SOLUTIONS

Recent trends, including more and better sensors, improved communication links and advances in data processing technology have produced more data and associated improvements in processing capability. However, large amounts of data cannot be assimilated rapidly enough for timely decision making. In addition, operational situations have become more complex and faster moving, creating the need for personnel to have expertise in multiple warfare areas and to make decisions in ever shorter time periods.

These developments, coupled with limited human cognitive processing capacity, necessitate the development of systems with a higher degree of integration and automation. The degree of integration and automation required can only be achieved by newly evolving technologies and techniques. The following three elements together may provide a technology solution to the data overload problem:

- artificial intelligence (AI) technology
- human-computer interface (HCI) technology
- decision augmentation orientation

AI technology will play a large role by providing automatic and augmented functional support. AI is a set of techniques that can be employed to develop systems that simulate human cognitive functions, such as problem solving, decision making and information processing. Advances in AI are occurring rapidly and by the 1990's, AI will be able to perform many decision functions currently performed by humans.

HCI technology deals with the methods by which a human and computer interact. It includes methods of information presentation, data entry, and function sequence control. By careful structuring of the interface and exploitation of innovative HCI techniques, such as color graphics, automated voice recognition and synthesis display windowing, etc., information transmission can be improved, and user learning time and processing demands reduced.

The proposed system design approach is facilitated by introducing decision augmentation. Decision augmentation software is designed to satisfy human information requirements given a particular mission and at some level perform the data to information transformation. Information is defined here as data processed to match the cognitive capability of the human and directed toward a specific decision task. Decision augmentation software is also designed to quantify information uncertainty and assist in using uncertainty in decision making. The decision augmentation orientation focuses on the decision requirements necessary to successfully accomplish mission objectives. Augmenting classical techniques with AI and HCI technologies can provide the means by which the proposed approach can be implemented in the operational environment.

A decision augmentation framework is most important when decision dependency in the system is high. The first graph in Figure 1 shows that theoretical system performance using classical approaches drops off as decision dependency increases. The second one shows that use of a decision augmentation approach should bring operational performance closer to its theoretical limit. Figure 2 shows a simplified schematic of a C^2 system with and without a decision augmentation orientation. In either case, data including intelligence, environment, sensor, threat, and own force, comes into the system and is processed. Without a decision augmentation framework, the processed data is pesented directly to the decision maker. The two shaded boxes are added with a decision augmentation framework. Combining decision augmentation software and innovative information presentation techniques, tailored to the specific decision-making task, should greatly improve decision-making quality. Decision augmentation software can include any or all of the following:

- "expert" knowledge bases and decision rules (AI technology)
- ability to structure the situation into a set of well-defined alternative courses of action
- capability to predict the consequences of each alternative course of action
- rank ordering of each alternative against one or more utility measures

Each of these, and other decision augmentation techniques, help to overcome some cognitive processing limitations and biases, and therefore improve the decision-making process. Humans have limited working memory and are not proficient at doing complex numerical calculations unaided. Furthermore, in evaluating alternatives, they often make simplifying assumptions, are biased toward considering solutions from their past experience, and usually do not simultaneously consider more than about three hypotheses. In general, when confronted with excess data, people resort to heuristics which simplify the problem and reduce the amount of data that must be considered. These heuristic biases may, in fact, be erroneous. Decision augmentation, particularly using AI techniques, allows more complex decision rules to be incorporated, more alternatives to be investigated and complex and accurate calculations to be made. This provides the ability to predict action consequences and evaluate their associated utility.

Information presentation, the second additional box, is based on an understanding of human cognition, and includes the following:

- fusion of sensor data, presenting only relevant data formated to directly support the decisions which must be made
- use of graphics, color and easy access of backup data (HCI) technology

Mechanisms of information presentation directly affect the ease with which the operator can assimilate and use the information displayed must be relevant to the operator's needs. The information should be structured, labeled and coded to highlight information content and relationships. Graphic displays are superior to alphanumeric displays for representing overall relationships among variables, with alphanumeric displays being most appropriate when precise information on specific variables is required. Color is a powerful highlighting technique and can be utilized to draw attention to the most important aspects of a situation. Since the displays should present only relevant information, easy access of backup data should be provided, including specific sensor data to resolve conflicts, or historical data to analyze specific trends.

LEVELS AND TYPES OF DECISION AUGMENTATION

Decision augmentation systems vary in the level of automation involved. They range from completely automatic, in which no operator action is involved, to manual, in which only computational aid is provided for the decision maker. Intermediate levels of decision augmentation include semiautomatic, in which the system provides decision alternatives and recommended courses of action and the operator reviews and accepts or overrides the system, and interactive, in which the operator and software support each other symbiotically. Level of decision augmentation is determined based on, at a minimum, the following considerations:

- operational situation
- data load
- decision importance
- intuition likelihood
- user acceptance (with decision level)

One of the major operational situation factors influencing decision level is time in which a decision must be made. Time to decide is only one variable in the time domain. Three time variables must be minimized to control the operational situation.

$$T_{control} = T_{collect} + T_{decide} + T_{transmit}$$

In order to maintain operational control:

$$T_{control} < T_{crit} - T_{op}$$

where: T_{crit} = critical time (time within which the operation must be executed in order to have the intended effect)

T_{op} = time required to execute the operation

If time were the only factor impacting decision level, their inverse relationship would suggest that the less time available for decision making, the higher the level of autonomy.

Even though time should be considered the most important factor when determining which decision augmentation level to use, all factors listed (and perhaps others) must be considered. Table 1 shows the relationship between the five independent factors and decision augmentation (DA) level in terms of an idealistic environment. It is easy to envision in a realistic environment that conflict among the factors is not only possible but highly likely, e.g., between operational environment and decision importance. If the decision is to have its intended impact, it may be necessary to have decisions made automatically when the critical time is very short, e.g., acitivate SDI to maximize boost phase kill, a very important decision.

Table 1

LEVEL OF DA	OPERATIONAL ENVIRONMENTAL	DATA LOAD	DECISION IMPORTANCE	USER ACCEPT	INTUITION LIKELIHOOD
Automatic	$T_{CRIT} = E$	High	Low	*	Low
Semi-Auto	$T_{CRIT} > E$	High	Mod	*	Low
Interactive	$T_{CRIT} \gg E$	Mod	Mod	*	Mod
Manual	$T_{CRIT} \ggg E$	Low	High	*	High

E = Short Time
* Variable, a Function of User

The level of decision augmentation desired determines which techniques should be used in implementation. Figure 3 shows where in the requirements analysis process this determination should be made and how it influences information processing and the HCI requirements. The type of decision augmentation techniques should be chosen to match decision situation requirements. While AI (expert system) techniques will prove extremely valuable, they are not applicable to all situations and should not be considered a panacea. A number of taxonomies for decision situations and decision aiding techniques have been proposed (Keen and Scott-Morton, 1978, Rouse, 1984; Wohl, 1981; Zachary, 1986) which may prove useful in determining the type of decision augmentation technique to use. Zachary suggests a taxonomy for decision augmentation based on the kinds of congitive support that the various computational techniques provide to human decision makers. For example, deterministic or stochastic process models support the selection of an action from a set of known alternatives by projecting the implications of each alternative based on assumptions about the process. In order to support reasoning processes in ill-structured problems with incomplete or contradictory data, AI techniques can be employed to provide symbolic reasoning capabilities based on a body of knowledge and specific kinds of inferencing procedures. Representational aids such as pictorial or spatial representations help the decision maker develop an understanding of a complex situation. Database management tools allow the user access to subsets of complex data aggregated according to a number of predefined or ad hoc criteria. Other types of tools support other decision making needs.

Working with the area of tactical command and control, Wohl presents a decision aid taxonomy based on the anatomy of tactical decision processes, called the SHOR model. The SHOR model defines four elements of a decision: Stimulus (data), Hypothesis (perception alternatives), Option (response alternatives), and Response (action). Information processing techniques are identified which are appropriate to each part of the decision process, depending on processing complexity, the time available

for the decision and the degree of information aggregation required. Some processing aids suggested include: Sensor correlation aids, zoom in/out with variable detail, speeded-up play back of selected battlefield history by target or unit type, knowledge/rule based systems, if/then triggers, English language data base access and pattern recognition aids, among others.

DECISION-ORIENTED SYSTEM DESIGN

The usability of a weapon or C^2 system is ultimately a reflection of its design philosophy. The most commonly applied system design approach begins with a detailed statement of platform mission requirements. Mission requirements are then used to derive functional requirements that will support successful mission accomplishment. The functional requirements are allocated to either hardware of software elements of the system. In practice, this approach emphasizes hardware considerations, with software being designed to facilitate hardware use.

Because informed, timely and "correct" decisions are the key element in system effectiveness, system design should be based on decision requirements. System hardware and software should be specified and designed to support the decision making function. Requirements should be based on operational performance deficiencies rather than on advanced technology, which is the tendency in a hardware oriented design approach.

Figure 4 defines an R&D approach which can be systematically applied to develop a C^2 system (or a weapon system) on a decision-oriented basis. This design approach also begins with the specification of mission requirements. Subsequent steps attempt to develop more specific system and subsystem functions to fulfill the primary mission requirements, as is currently done. Based on the mission analysis, all decisions necessary to perform specified mission functions are defined. The step of defining decision requirements is done early in the requirements analysis, and serves as the determinant of all hardware and software requirements. After the decision requirements are specified, each decision is analyzed to identify the information needed to make the decision. "Information" implies data that has been processed and reduced to just the elements needed for the decision. Next, the data necessary to provide decision-specific information is defined. "Data" refers to data, (e.g., target contact reports) that is needed to derive decision-specific information. These three steps are critical because they serve as the basis for developing all detailed requirements in the system specification.

Once the data necessary to provide decision-specific information have been defined, the hardware and software (both decision augmentation and other support software) requirements are specified. As Figure 4 shows, there are three parallel, but not independent, paths for specification of decision augmentation software, support software and hardware requirements. The emphasis on the decision function suggests that the middle path, that of defining decision augmentation software, is the leading path. First, it is necessary to identify the source of each data element. Then, decision functions and their information requirements must be allocated to organizational units/individuals and subsequently to human (specifc organizational units/individual) versus computer (decision augmentation software). The human/computer allocation should be based on the relevant capabilities and limitations of each. Decision augmentation requirements, including HCI requirements, should then be derived based on an analysis of the decision problem and the techniques to assist that particular decision problem.

The left-hand path, that of defining support software requirements, is based on the data necessary to provide decision-specific information as well as the decision augmentation software requirements. The support software might include operating systems, device handlers and data base management systems.

The design approach shows hardware requirements, the right-hand path, also being dependent on the "Define Data" task. First, the data parameters are defined. Hardware performance requirements are stipulated based on the hardware's capability to provide the data parameters (or performance) specified. This, then, determines the hardware specification, i.e., the ability to provide the data needed to produce information required for decision making. The hardware elements include sensors, which must furnish specified data at a certain rate and accuracy to allow a meaningful and timely decision to be made; computers which must have the requisite capacity and computational speed; and communciations systems which must have the required connectivity and bandwidth to handle data transmission load. Developing hardware requirements based on the decision maker's needs will provide a better fusion between these system components, resulting in a higher level of performance than previously achieved by using the current system design approach.

After the hardware, support software and decision augmentation specifications have been defined, the hardware/software interface specification can be defined. By combining these specifications appropriately, the overall system specificatin can be developed. What will result is a C^2 system (or weapon system) that has a greater likelihood of meeting its theoretical performance level.

ORGANIZATION ANALYSIS

Reorientation of the system design approach to emphasize the human and his decision making responsibilities requires analyses that are not considered in the current approach. These analyses are qualitative and concentrate on factors that contribute to the framework of decision making in an organizational environment. The analysis methods and data sources for the initial requirements analysis are shown in Figure 5. Mission requirements are determined by a functional analysis of operational problems and deficiencies, including tactics, sensor utilization, sensor performance, available resources, and enemy order of battle. Decision requirements are identified by going directly to the decision makers. Non-quantitative behavioral/social science methods such as questionnaires, interviews, verbal protocols and observation of the decision-maker in his operatinal environment should be employed. Analysis of the operational and organizational environment is also crucial. It should include identification of the following elements:

- informal as well as formal communication links
- apportionment of decision-making responsiblilty to components of an organization
- relationships between organizations
- what decision aids (automated or not) are currently being used or could be used if available.

Analysis of these and other organizational elements will enhance the ability to correctly define decision requirements and to determine the appropriate level of decision augmentation.

TECHNOLOGY IMPLICATIONS

Emphasis on decision making functions has a number of implications for the design and implementation of C^2 systems (and weapon systems). Five are discussed below.

First, a basic understanding of human cognitive processing is required to fully realize decision augmentation potential. C^2 systems should be designed to provide assistance in those areas where human capability is limited while still capitalizing on human strengths. For example, humans have attention and memory limitations, inherent heuristics which they employ in information processing and limmited ability to process numerical data in complex, stressful and data overload situations. These are areas in which decision augmentation can provide significant improvements over unaided decision making.

Second, HCI design can affect decision behavior; therefore, its effects should be considered in system design. Decision augmentation system design is often concerned with methodological validity, without sufficient attention to the relationship between it and the user (e.g., the amount and kind of user-augmentation interaction, dialogue style, information presentation formats, Schwartz and Jamar, 1983). If results of decision augmentation are not presented in a directly useable format with due consideration to the user's needs and desires, they will not be effecitvely used. Importance of the interface between human and computer has received increased attention in recent years and is particularly critical in situations in which decisions must be made under time pressure or conditions of high data volume. Some examples of good interface design features are:

- use of graphics to represent situational overviews, particularly geographic representations
- provide only that information needed to support a decision situation
- display historical data on request
- provide embedded training and on-line tutorials to facilitate use by both novices and experts
- provide easy means of user-computer communication
- make the knowledge base and decision rules in decision augmentation systems accessible so the user can query them
- insure computer response speeds are commensurate with user expectations
- provide automatic mode settings that users can override (e.g., number of alternatives display, what utility measure to order alternatives on)
- provide suggestive rather than authoritative output
- provide succinct rather than conversational output

Third, improved understanding of and attention to innovation acceptance is needed. Whether a decision system is used or not will depend not only on its design but also how it is introduced (Mackie and Wylie, 1985). The system must be designed so that it meets the user's needs rather than introducing additional workload. Even if the system is actively involved in the decision-making process, and in some cases excludes the human, it should not appear to erode individual control or decision making authority. System operation should require only minmum knowledge of computers and should not involve complex operating procedures. The user community should be involved throughout the development process to facilitate the introduction of new decision automation systems (Adelman, 1982). They should be involved early in the development cycle, when system requirements analysis is being conducted. Also, prior to system introduction, potential users should be briefed on system capabilities and operating procedures so they know what to expect and can take full advantage of what is provided.

Fourth, AI technology is rapidly becoming accepted as a major tool for implementing decision augmentation systems. It allows the use of more complex decision rules in addition to numerical computations and algorithms. Furthermore, the knowledge of experts can be acquired and encoded into the system knowledge base.

Finally, decision augmentation systems should be able to adapt to both user and environment. In order to make the system adaptable, the system architecture in which the knowledge base and decision rules reside must permit change. It should be possible to update when new tactical situations arise or other environmental changes occur. Also, there are individual differences among decision makers both in decision making style and the importance they place on different criteria in determining a final

decision. This individuality should be accommodated to the extent possible using innovative architecture until technology has evolved to the point where learning can be an integral part of the decision augmentation system. The current system design approach does not bring into focus the technology implications discussed. It is only through a decision oriented design approach that full advantage can be made of the new technology.

SUMMARY

As a consequence of the increased data volume in current and future C^2 systems and the limitations in human cognitive processing capacity, a reorientation of the system design process has been proposed. In the approach proposed, system design is based on decision requirements rather than on hardware performance. Also, decision augmentation techniques and innovative information presentation are needed to reduce the data overload. The higher degree of integration and automation possible with decision augmentation systems coupled with the emphasis on decision functions should greatly reduce the incoming data load so that the decision maker can devote more time to thinking about operational problems rather than merely reacting to the task environment. If these changes can be accomplished, C^2 system capabilities should improve, and as a consequence, increased mission effectiveness.

REFERENCES

Adelman, L. Involving users in the development of decision-analytic aids: the principle factor in successful implementation. Journal of the Operational Research Society, 1982, 33, 333-342.

Keen, P.G. W. & Scott-Morton, M.S. Decision Support Systems: An Organization Perspective, Reading, MA: Addison Wesley, 1978

Mackie, R. R. & Wylie C. D. Technology Transfer and Artificial Intelligence: User Considerations in The Acceptance and Use of AI Decision Aids. Goleta, CA: Human Factors Research Division Essex Corporation, Technical Report TR51231-1, November 1985

Rouse, W. B. Design and evaluation of computer-based decision support systems. In S. J. Andriole (Ed.), Microcomputer Decision Support Systems. Wellesley, MA: QED Information Sciences, 1984.

Schwartz, J. P. & Jamar, P. Lack of guidance for decision aid interface design. Association for Computing Machinery SIGCHI Bulletin, 1983, 15, 13-17.

Wohl, J. G. Force management decision requirements for Air Force Tactical Command and Control. IEEE Transactions on Systems, Man, and Cybernetics, 1981, SMC-11, 618-639.

Zachary, W. A cognitive based functional taxonomy of decision support techniques. Human-Computer Interaction, 1986, 2, 25-63.

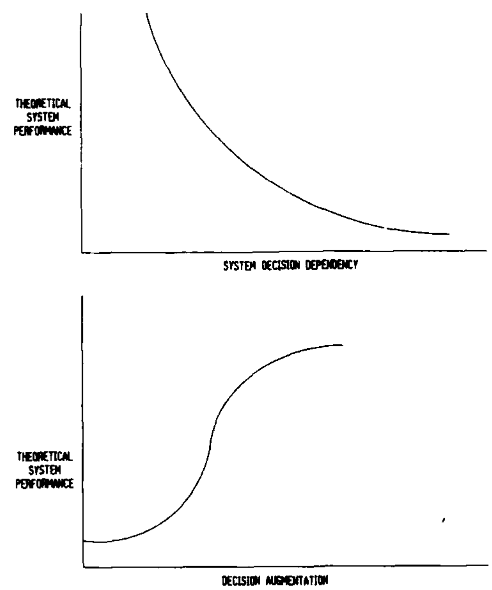


FIGURE 1. SYSTEM PERFORMANCE DEPENDENCIES

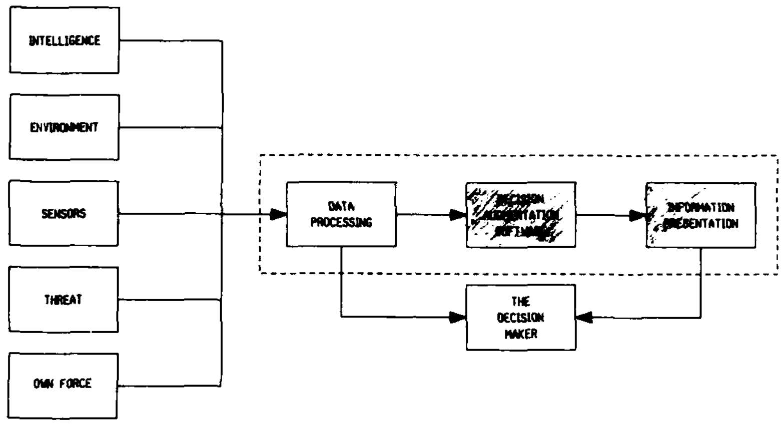


FIGURE 2. DECISION FUNCTION EMPHASIS

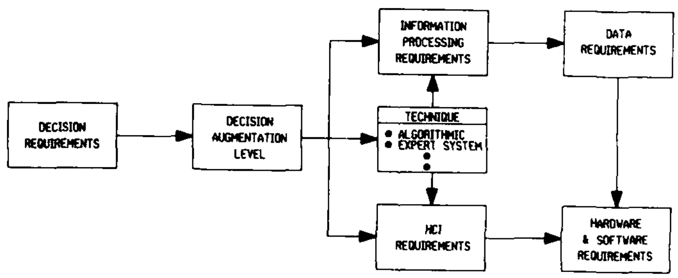


FIGURE 3. IMPACT OF DECISION AUGMENTATION LEVEL

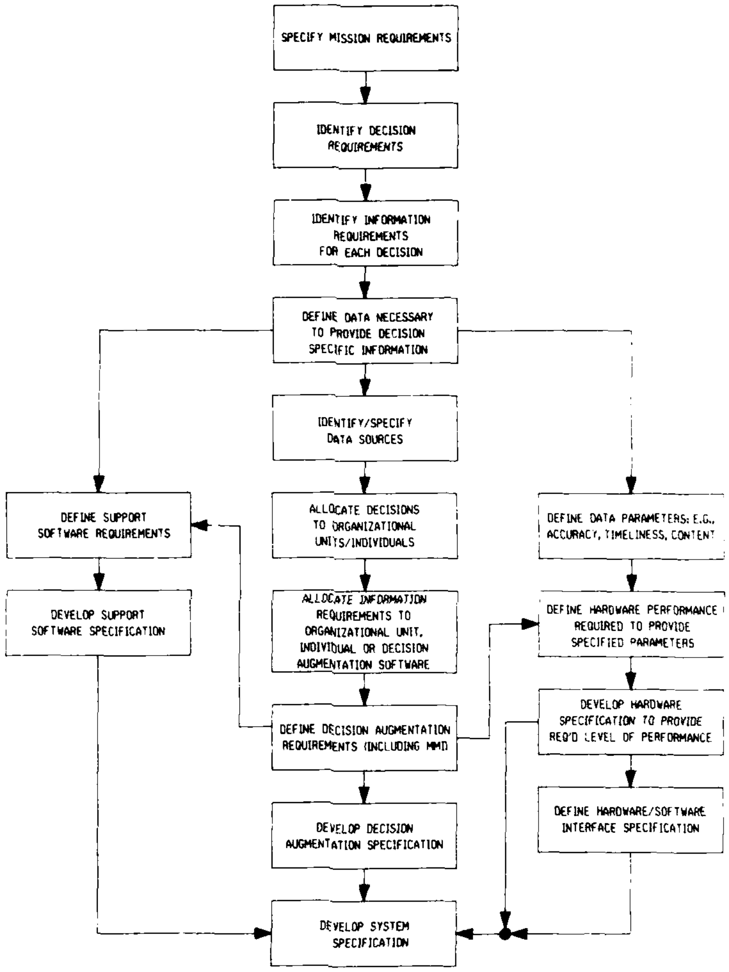


FIGURE 4. DECISION-ORIENTED SYSTEM DESIGN

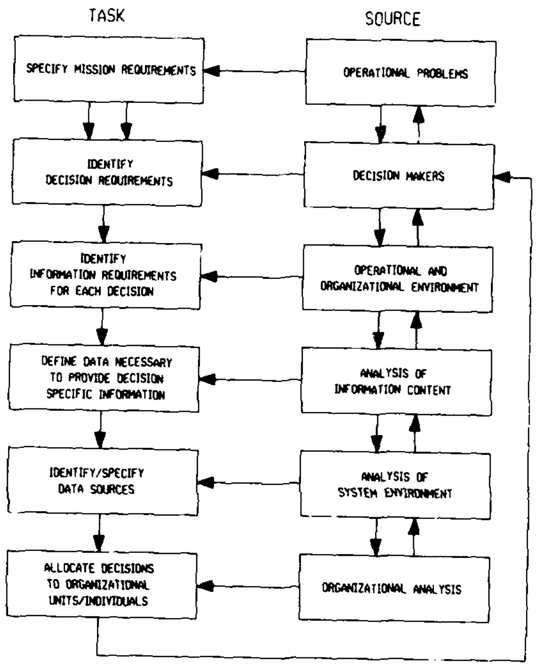


FIGURE 5. ANALYSIS METHODOLOGY

Managing Advanced Avionic System Design

Philip H. Simons
Project Engineer, ASET Development
Northrop Corporation
One Northrop Avenue
Hawthorne, CA 90250

Summary

The application of new technologies such as VHSIC, and new system concepts such as artificial intelligence, distributed operating systems, common hardware components, and reusable software modules/libraries has changed the manner in which avionics suites are designed, developed, and verified. The requirements for developing these systems are specified in U.S. military standards and can be segregated into requirements for the overall system, configuration items, interfaces, and the design process management. Northrop has developed a design process and methodology capable of meeting these system development requirements which involves time phased analyses, requirements analysis and decomposition, functional decomposition, and system synthesis and interface definition. This process, developed through past advanced avionic system design efforts, is now being automated through the development of a set of tools to manage all phases of advanced avionic system design, the Avionic System Engineering Tool (ASET).

I. Advanced Avionic System Design Requirements

The design of any system includes key elements such as software, hardware, and interfaces. Although the requirements for these elements can be specified in any number of ways, the U.S. Department of Defense (DoD) has mandated a specific set of documents to contain these requirements. These documents are defined in MIL-STD-483, MIL-STD-490, and DoD-STD-2167, and include a description of the following requirements for advanced avionic system design:

1. Overall system requirements. These requirements include the following:

    a. Definitions of the major functions of the system, and the principal interfaces between those functions

    b. The allocation of performance requirements and specific design constraints peculiar to each system function

    c. The definition of the principal interfaces between the system being specified and other systems with which it must be compatible

    d. The operating requirements for logistically and technically supporting the system

    e. The design requirements necessary to assure compatibility of system equipment and software

    f. The identification and relationship of Hardware Configuration Items (HWCIs) and Computer Software Configuration Items (CSCIs) which comprise the system.

2. Configuration item requirements. These requirements include the following:

    a. The detailed functional, performance, interface, and qualification requirements of all Configuration Items (CIs)

b. The identification and relationship of the components of all CIs

c. The requirements for programming design, adaptation, quality factors, and traceability of all CIs

d. The identification of the subset of the overall system requirements allocated to each CI.

3. Interface requirements. These requirements include the following:

a. The detailed requirements of the interfaces between all components of the system

b. The methods by which these interfaces will be verified.

4. Design process management requirements. These requirements include the following:

a. Plans for the development, test, configuration management, and quality assurance of the system and the CIs within the system

b. Documents describing the procedures to be followed during the development, test, configuration management, and quality assurance of the system

c. Documents describing the procedures and information necessary to maintain the system after it is passed on to a customer.

These requirements must be generated during the design of any system. Section II will detail the process by which Northrop has satisfied these requirements in the past, and Section III illustrates how this process is being automated to improve the accuracy and efficiency of Northrop's advanced avionic system designs.

II. Advanced Avionic System Design Process

The avionic system design process traditionally consists of four steps, repeated iteratively until a satisfactory system design is accomplished.

1. Abstract requirements identification. The process of requirements identification involves determining the high level requirements of the avionic system. These abstract requirements are derived from mission phase decompositions and system timeline analyses, or can be extracted directly from Statement of Work (SOW), Contract requirements, System Requirements Document (SRD), or Weapon System Specification (WSS).

2. Requirements/functional decomposition. The requirements arrived at during the abstract requirements identification phase of system design are usually not those directly usable in the design of a system. Prior to performing actual system design, these requirements must be refined. For example, a requirement such as "The system must be capable of displaying a target of 20 m^2 at 25 nm" could be refined by the systems engineer into the following:

a. "There must exist a sensor capable of detecting a target of 20 m^2 at 25 nm."

b. "There must be a display system capable of showing a target to the pilot with a range of 25 nm and 20 m^2."

c. "There must be a communications path between the sensor and display system called out above."

This procedure of refining requirements continues until the process phases into a functional description and decomposition procedure. This point is reached when the systems engineer describes a function which performs the requirements, rather than simply describing the requirements in increasingly greater detail. An example of this would be the definition of a Head-Up Display (HUD) in order to meet a portion of the requirements of the display system listed above.

This process is also characterized by decomposing and refining requirements or functions from different viewpoints. In other words, the system can be described from the viewpoint of non-combat-flight vs. air-to-air vs. air-to-ground, or offensive vs. defensive, or even from the viewpoint of the hardware system as it is currently defined. Examining the system from these disparate viewpoints will often identify requirements or functions not illustrated during the original decomposition. It must be noted that the same lower level requirement or function can be arrived at from different higher level entities, i.e., the HUD specified above would meet the requirements for the display of many items, not only the target at 25 nm. Another aspect of this phase of system design is the assignment of functional inputs, outputs, estimates of memory, throughput, size, weight, and power.

Finally, the result of this phase of system design is the detailed description of a large set of functions which must be performed by the avionic system.

3. Functional recomposition. The functional recomposition process involves taking the large set of functions arrived at during the requirements/functional decomposition phase of the system design, and grouping these functions into increasingly larger sets. For example, all functions dealing with the display of targets in air-to-air mode might be grouped together. Then this set could be included in the set containing air-to-ground target display functions, to form an all-mode-target display group. This process continues until all functions have been included into groups, all of which have been grouped into the complete system.

   This functional recomposition is performed by the system engineer, and requires forethought and research prior to performing the groupings. The tasks the engineer is accomplishing in this phase of system design include the following:

   a. The definition of the system elements, from the most detailed descriptions of functions to the actual Line Replaceable Units/ Line Replaceable Modules (LRUs/LRMs)

   b. The assignment of signal paths for all input/output pairs within each group, as that group is created

   c. The summation of the estimates (memory, throughput, size, weight, etc.) of the lower level functions into estimates of the high level groupings.

   The end result of the functional recomposition phase is the actual system design, minus the detailed interface definitions.

4. Detailed interface definition. During the detailed interface/bus definition phase of system design, the engineer first defines and then assigns the information associated with each signal path within the system. In other words, the engineer chooses the communications path (i.e., bus, memory link, etc.) to be used for a given signal path, defines the information specific to that communications path, and then assigns the bus-specific information.

   During the requirements/functional breakdown phase of the system design, an engineer assigns inputs and outputs to the functions within the system. These signal definitions include information such as frequency of update, units, maximum/minimum value, etc. These definitions do not include information specific to a certain bus, such as remote terminal number, message number, bit position, etc. Not only is this information bus-specific, but the task of assigning this information during the requirements/decomposition phase is extremely complex, and requires very good communications between different engineers. After all signals traveling on a path are defined, however, the task becomes greatly simplified. Therefore, the mapping of the signals onto the bus-specific layout is accomplished after functional recomposition. after the signals are defined and specified for each bus.

   This four-step process is repeated many times, with the end result being the system design. This system design is captured in documents defined by the MIL-STD and DoD specifications, and fulfills the requirements of the system design as defined in Section I. The correlation between these requirements and the system design phases is shown in Figure 1.

III. Automated Tools for System Design

Automated tools can greatly aid in the avionic system design process, providing more efficient, more easily managed, and more accurate system designs. For these tools to be truly usable and effective, however, the following requirements must be followed:

1. Information regarding tools on the system must be disseminated to all potential users, not only those specifically targeted for tool use.

2. The tools and systems must be capable of expanding and accommodating new tools, migrating to new systems, accepting larger and more elaborate problems and, in general, supporting the future requirements of avionics engineering as well as the current needs. Much of this requirement has been fulfilled simply by documenting to MIL-STD-2167, coding in a Higher Order Language (HOL), and making full use of the HOL's built-in modularity capabilities.

3. The tools must be able to be quickly and efficiently integrated into the existing Avionics Engineering environment. This involves the utilization of currently existing hardware (terminals and mainframes), as well as maintaining a user interface very similar to existing tools.

4. The tools must be both powerful and user-friendly. The most involved on-line queries must take less than 30 seconds; typical operations must take far less. In addition, the tools must include extensive help utilities.

SYSTEM DESIGN PHASE	SYSTEM DESIGN REQUIREMENTS FULFILLMENT
ABSTRACT REQUIREMENTS IDENTIFICATION	OVERALL SYSTEM REQUIREMENTS CONFIGURATION ITEM REQUIREMENTS DESIGN PROCESS MANAGEMENT REQUIREMENTS
REQUIREMENTS/ FUNCTIONAL DECOMPOSITION	OVERALL SYSTEM REQUIREMENTS CONFIGURATION ITEM REQUIREMENTS
FUNCTIONAL RECOMPOSITION	OVERALL SYSTEM REQUIREMENTS CONFIGURATION ITEM REQUIREMENTS
DETAILED INTERFACE/BUS DEFINITION	INTERFACE REQUIREMENTS

431 007 87

FIGURE 1. REQUIREMENTS FULFILLMENT BY DESIGN PHASE

5. The tools must not require a long learning curve to achieve initial benefits.
6. The tools must be capable of providing hardcopy output for review by others (such as subcontractors).
7. The tools must facilitate securing any information entered into the system.

Northrop's Aircraft Division is developing an automated tool called the Avionic System Engineering Tool (ASET). The ASET fulfills all of the just-listed requirements, and is intended to provide engineers the ability to perform the end-to-end system design process more quickly and effectively, while improving both the turnaround time and accuracy of the documents associated with the system. The ASET will also provide traceability from all levels of requirements through the actual system hardware and software CIs.

Past system design efforts have relied on paper-based documentation which was often out-of-date, difficult to understand, internally inconsistent, and usually did not reflect the entire avionic system. As a result, the test engineers, software developers, subcontractors, implementors, etc., had to rely on conversations with the original designer of any system to determine exactly what had been specified.

Another difficulty of the paper-based documentation scheme is the separation of the engineers from their output. The design engineers prepare their work on paper, which is redone by either their secretary or drafting personnel for placement in the document. The personnel requiring this information must then acquire a copy made of the document.

The manner in which the ASET overcomes many of the difficulties of the paper-based documentation effort is by involving the avionic system designers themselves in the process of generating the system representation. This does not mean that they have to involve themselves in standard data entry functions. The ASET allows the engineers to perform their jobs in the same manner that they have always performed their jobs, only faster, as if they had an intelligent piece of paper and pencil; ASET is recording their work and saving it for later steps in the process. Thus, ASET improves upon the

system engineering process by allowing the design engineers to manipulate and create the actual system representation accessed by the personnel requiring this information.

The ASET has been divided into four subtools – the Timeline Analyzer Subtool (TAS), the System Analysis Subtool (SAS), the System Generation Subtool (SGS), and the Interface Definition Subtool (IDS), which correspond to each phase of the system design process, as shown in Figure 2.

SYSTEM DESIGN PHASE	SYSTEM DESIGN REQUIREMENTS FULFILLMENT	CORRESPONDING ASET SUBTOOL
ABSTRACT REQUIREMENTS IDENTIFICATION	OVERALL SYSTEM REQUIREMENTS CONFIGURATION ITEM REQUIREMENTS DESIGN PROCESS MANAGEMENT REQUIREMENTS	TIMELINE ANALYSIS SUBTOOL (TAS)
REQUIREMENTS/ FUNCTIONAL DECOMPOSITION	OVERALL SYSTEM REQUIREMENTS CONFIGURATION ITEM REQUIREMENTS	SYSTEM ANALYSIS SUBTOOL (SAS)
FUNCTIONAL RECOMPOSITION	OVERALL SYSTEM REQUIREMENTS CONFIGURATION ITEM REQUIREMENTS	SYSTEM GENERATION SUBTOOL (SGS)
DETAILED INTERFACE/BUS DEFINITION	INTERFACE REQUIREMENTS	INTERFACE DEFINITION SUBTOOL (IDS)

43100887

FIGURE 2. ASET SUBTOOL CORRELATION MATRIX

The TAS allows the engineer to enter mission phases into a timeline, and then decompose any phase of that mission into a complete, lower level timeline. In addition, the TAS provides full cut-and-paste capabilities, and is able to pass the requirements entered by the engineer onto the System Analysis Subtool.

The SAS provides the capability to perform requirements/functional decomposition on the CRT, alternating between displaying the requirements tree as it is created, or displaying the requirements/functions and the entities above and below them in the tree. The SAS provides signal verification, verifies hierarchical consistency, and provides for both redundancy and nebulous requirements fulfillment checking. The SAS provides cut-and-paste functions on the tree, and also generates the set of functions passed on to the System Generation Subtool.

The SGS allows the engineer to perform the functional recomposition of the system, collecting like functions upon command, and allowing the engineer to specify which of the functions within the collection to group. In addition, the SGS performs I/O verification, allows assignment of signals to signal paths, and rebuilds the memory, throughput, size, weight, etc., estimates for each grouping within the system. The SGS builds the database which is used by the Interface Definition Subtool.

The IDS allows the user to define and then enter all bus-specific information associated with each signal in the database. The IDS also provides the capability to generate documents based on the information entered by the user such as the Interface Requirements Specifications and Sections 3 and 4 of the System Requirements Specifications, and is capable of providing information to CADAM systems for the generation of system layouts, wiring diagrams, etc.

IV. ASET Benefits

1. Requirements Traceability. The ASET permits the users to trace from any requirement or function within the system (entered or generated in the SAS), to the final avionic system or systems that fulfill that requirement. This permits anyone to determine exactly what requirements are fulfilled by any particular LRU/LRM, and to determine exactly which LRUs/LRMs fulfill any particular requirement.

2. Contract Modification Response Improvement. Due to the requirements traceability of the ASET, the impact of any changes to system requirements can immediately be determined.

3. I/O Verification. The ASET verifies that all signals within the system (input through the SAS) have constructs which generate, transfer, and receive these signals.

4. Throughput and Memory Analysis. The ASET requests the systems engineer to input memory and throughput estimates for each bottom level function within the SAS. When the system grouping is being accomplished, these memory and throughput estimates are combined to give estimates for any subset of, or the entire, avionics system.

5. Improved Inter- and Intra-Group Communications. Communications between different groups will improve due to the better system description being available. Communications within the systems engineering group will improve because engineers will be able to "try out" their design with the other tentative designs of the system. This will result in an even faster and more effective system design process.

6. Improved Software Design and Test. With the development of a better designed and documented avionics system, the software design and software test personnel will perform more efficiently, which will result in a more reliable, effective, and error-free system.

In addition to all that has been specified, the ASET has been modularly designed to easily accommodate enhancements such as the following:

1. Automatic code generation for portions of the system. One of the first enhancements planned for the ASET is the association of a Program Design Language (PDL), the implementation of which may be Ada, with each element in the SAS. This defines sequentially executed operational steps for each function within the system. This PDL can then be used to generate code.

2. System prototyping and analysis. With the association of a PDL with each SAS element, system prototyping becomes merely the execution of the PDL (or code generated from the PDL) based on the signal paths and system hierarchy defined in the SGS and IDS.

3. Bus loading analysis. As bus definitions are added through the use of the IDS, bus loading analysis packages can be developed for the different buses to aid in developing the interfaces within avionics systems.

V. Conclusion

In summary, the critical requirements of advanced avionic system design are overall system requirements, configuration item requirements, interface requirements, and design process management requirements. These requirements can be met through the design process of abstract requirements definition, requirements/functional decomposition, functional recomposition, and detailed interface definition. This process can be greatly aided by computer automation, resulting in the design of more complex avionics systems in far less time than would be possible using older tools.

References

1. MIL-STD-483, Configuration Management Practice for Systems, Equipment, Munitions, and Computer Software
2. MIL-STD-490A, Specification Practices
3. DoD-STD-2167, Defense System Software Development
4. DoD-HDBK-287. Defense System Software Development Handbook.

DISCUSSION

J.Nicol, UK

While recognizing the potential of "ASET" for new aircraft system design, how relevant would it be in a retrofit situation where the customer stipulates not only the requirement but also the hardware?

Author's Reply

The functional requirements decomposition phase of system design, as automated by "ASET," requires only that functions be decomposed to the point where each function resides in a separate box. While "ASET" supports further decomposition, it does not require it. Therefore, the engineer need only decompose requirements to sufficient detail, define the prespecified hardware to be the function that satisfies those requirements, and then define that function to be a leaf-level function.

ERGONOMIE PSYCHOSENSORIELLE DES COCKPITS, INTERET DES SYSTEMES INFORMATIQUES INTELLIGENTS

AMALBERTI R., Médecin-Principal, Spécialiste de Recherches *
DEBLON F., Ingénieur de l'Armement, Ingénieur de Recherches *
MENU J.P., Médecin-Principal; Spécialiste de Recherches *

* Centre d'Etudes et de Recherches de Médecine Aérospatiale
Laboratoire Central de Biologie Aérospatiale
5 bis, avenue de la Porte de Sèvres
75731 PARIS CEDEX 15 AIR

RESUME

L'ergonomie psychosensorielle des cockpits s'oriente de plus en plus vers des applications dynamiques interactives et non limitées à l'application de normes. Le but est de mieux répondre aux besoins cognitifs du pilote, de mieux l'aider dans sa tâche réelle. Les systèmes informatiques intelligents sont indispensables à cette évolution. Deux types d'applications sont décrites : la première consiste, en disposant de données précises sur la perception visuelle, à modifier l'image source pour y faciliter la vision de tel ou tel détail. La seconde consiste à élaborer un outil d'aide au pilotage préservant le pilote de toutes ses possibilités d'intervention et de décision mais minimisant ses défauts (fautes d'inattention, routines....) Un modèle opérateur obtenu par mise à plat de l'expertise d'un pilote de combat a été programmé dans ce cadre et sert à établir la faisabilité du projet.

ABSTRACT

Psychosensory cockpit ergonomics consists in a pluridisciplinary approach, focused, we believe, on global consideration of man-machine interface issues. Knowledge supplied by each research field (sensory physiology, cognitive psychology, design of intelligent systems) is used in a very concrete approach, taking into consideration aviation requirements and technological advances.

Intelligent computer systems ("intelligent" should be preferred to "expert") are ergonomics systems since they adapt the rigidity of current data processing systems to man- and situation-related variabilities.

To build such systems, certain prerequisites have to be met:

— know exactly the task performed by the user (set of rules)
— know exactly transfer functions of sensory organs,
— be able to define exactly physical stimulations.

Based upon this knowledge, the system can follow the mission as a function of its real course and pilot's strategies.

Ergonomics goals can be listed in a hierarchy:
— self-operating transparent system providing protection to man machine couple,
— transparent system, modifying psychophysical properties of displayed data to achieve best possible adaptation to sensory capabilities,
— consultant system for solving problem situations.

These various themes are discussed, using examples from work done in our department, which demonstrate the value of this approach.

INTRODUCTION

L'ergonomie psychosensorielle des cockpits a beaucoup progressé depuis la célèbre enquête de FITTS & JONES (1947) mais il s'agit toujours dans son objet scientifique de minimiser les problèmes posés par l'interface entre deux modes de fonctionnement différents : celui de l'homme et celui des machines. Comme l'homme est le directeur de ce couple, l'ergonomie vise à modifier l'interface afin de contraindre le moins possible ce partenaire humain à perdre de l'énergie et du temps dans la difficulté de communiquer (donc de réserver ses ressources à la tâche elle-même). L'ergonomie psychosensorielle concerne traditionnellement l'adaptation des sorties de la machine aux caractéristiques des entrées sensorielles de l'opérateur. Ainsi en va-t-il des règles de choix de taille de caractères, de couleurs, de contrastes, plus récemment de position des visualisations dans le cockpit. Des normes internationales de standardisation sont souvent disponibles et cette ergonomie statique s'appuie solidement sur un large "base" de connaissances théoriques acquise par la psychophysique. A ce jour, les résultats sont déjà grands et il est devenu difficile sinon impossible de concevoir un cockpit sans s'intéresser de près à ces éléments.

Depuis quelques années, cette ergonomie psychosensorielle a franchi un nouveau cap, prenant parfois le nom d'ergonomie cognitive. Elle voudrait privilégier maintenant une adaptation de l'interface non plus limitée aux seuls aspects statiques de la communication (normes visuelles, auditives, tactiles...) mais étendue aux aspects dynamiques de la prise et du traitement des informations ; modifications des caractéristiques psychophysiques sous contraintes aéronautiques ; besoins informationnels, raisonnements, stratégies .... En un mot, elle voudrait être plus près de la tâche journalière du pilote.

Le développement parallèle des techniques informatiques rend possible cette prise en compte dynamique. L'interface possède alors une certaine souplesse et c'est cette souplesse que l'on peut qualifier de doublement intelligente : par sa nature technologique faisant appel à l'intelligence artificielle, par son but mettant en jeu l'idée de flexibilité intelligente des afficheurs en fonction des besoins de l'usager.

Ce long point introductif doit inclure la position originale de ce concept d'aide intelligente dans les cockpits par rapport aux mêmes concepts habituellement développés par les bureaux ingénieurs (en précisant que ces conceptions, loin d'être exclusives sont complémentaires).

Rappelons avec BISSERET (1983) qu'il existe deux grandes classes de systèmes d'assistances : ceux crées pour remplacer l'opérateur dans certaines tâches (stratégie de l'automatisation) et ceux crées pour être des aides interactives à l'opérateur (stratégie de l'aide personnalisée). La démarche des ingénieurs se situe souvent dans la première perspective ; la démarche du laboratoire se situe fondamentalement dans la seconde perspective. Elle s'organise selon deux applications essentielles prenant leur source dans des concepts émanant pour l'un plutôt de la physiologie sensorielle, pour l'autre plutôt de la psychologie cognitive sans qu'il y ait (à nouveau) vraiment d'exclusive entre ces approches. Voici ces deux concepts qui vont organiser le plan du texte :

- par la connaissance des modèles perceptifs, il est possible de concevoir un asservissement temps réel de l'affichage (dans ses qualités psychophysiques) afin qu'il facilite à l'opérateur le prélèvement et le traitement des informations. Cette possibilité peut s'étendre à la possibilité de flexibilité du contenu informationnel des afficheurs en fonction des besoins.

- par la connaissance des modèles mentaux utilisés par le pilote (raisonnements, anticipations, cohérences), il est possible d'aider le pilote dans la gestion et la surveillance de paramètres qui ne sont pas directement impliqués dans le cours d'action. Ce concept est finalement assez proche de l'idée de WIENER et CURRY (1985) de pilotage par transparence car le système informatique ne se manifeste qu'aux limites du domaine de sécurité du vol.

Ces deux programmes sont en cours de développement au C.E.R.M.A.. Ils relèvent encore pour de nombreux aspects de la phase de recherche car ils nécessitent la construction préalable de modèles de fonctionnement cognitifs du pilote et leurs mise en jeu dans des systèmes informatiques de maîtrise architecturale encore délicate.

I - MODELES DE LA PERCEPTION ET FLEXIBILITE DES AFFICHEURS

1.1. Bases théoriques :

Deux types de travaux constituent la base d'un certain modèle perceptif fonctionnel.

- le premier concerne la modélisation de la fonction de transfert des organes sensoriels, essentiellement la vision dans le cadre de cette étude.

Tous les systèmes sensoriels ne sont sensibles qu'à certaines classes de stimulations et à l'intérieur de ces classes à seulement une partie des stimulations possibles. On peut étudier, en analogie avec la fonction de transmittance d'un système quelconque, la fonction qui décrit au mieux le filtre imposé par cet organe sensoriel aux stimulations du monde physique. Une telle relation est appellée "fonction de transfert" de l'organe sensoriel considéré. En vision, les stimuli sont des radiations électromagnétiques ; l'oeil n'est sensible qu'à un intervalle étroit de valeurs de ces radiations : le spectre visible (400-760 nm). A l'intérieur de ce spectre visible, la fonction de transmittance se décrit dansle domaine spatial et dans le domaine temporel. CAMPBELL et GREEN (1965) ont montré que la transmittance du système visuel dans le domaine spatial est caractérisée par une fonction de sensibilité aux contrastes spatiaux. MENU (1985) a précisé la valeur standard de cette fonction pour les trois couleurs fondamentales, en central et en excentricité jusqu'à 40°. L'oeil se présente comme un filtre passe bande haut et passe bande bas (figures 1).

Cette fonction se modifie sous l'effet de facteurs d' agressions aéronautiques. L'hypoxie a été le premier facteur testé. 13 sujets jeunes ont participé à une expérimentation de recueil des FSC polychromes à plusieurs niveaux d'hypoxie correspondant à des altitudes simulées en caisson à dépression de 3.500, 4.500 et 5.500 m. (F.S.C. : Fonction de Sensibilité aux Contrastes, F.S. : Fréquences Spatiales)

L'hypoxie modifie les seuils des différentes fréquences spatiales en fonction des 3 couleurs primaires testées. Les seuils des hautes fréquences spatiales en bleu sont les premiers modifiés. L'atteinte est d'autant plus grande que l'altitude est élevée et elle gagne les F.S. moyennes. Les seuils des F.S. basses sont les mieux conservés. Pour le rouge et le vert les seuils sont moins perturbés surtout pour les F.S. élevées. Le vert est la couleur dont les seuils sont les moins modifiés. MENU (1986) a pu établir la banque de données permettant d'esquisser un modèle fiable d'altération pour cette agression.

Des résultats comparables sont également acquis en ce qui concerne l'agression lumineuse (intensité lumineuse, taille du champ de stimulation).

Les travaux se poursuivent par contre au C.E.R.M.A. pour établir la même banque de données pour les accélérations et les vibrations.

- la seconde concerne les étapes ultérieures de codage de l'information. Au-delà du premier filtrage, des résultats très importants ont été obtenus en ce qui concerne la vitesse de transmission des informations en fonction de la fréquence spatiale sous laquelle elles sont présentées. Les détails les plus larges nécessitent les temps de transmission les plus courts. Deux systèmes anatomiques et neurophysiologiques différenciés sont responsables de cet écart : les voies des cellules ganglionnaires de type X et des cellules ganglionnaires de type Y. Les cellules de type X ont des champs récepteurs beaucoup plus petits que ceux des cellules Y, leurs champs dendritiques sont limités. La vitesse de conduction des influx nerveux dans les axones est plus lente pour les cellules de type X par rapport aux cellules de type Y. Les cellules de type X correspondent à des photorécepteurs situés dans la zone de vision centrale. Les cellules Y correspondent à des photorécepteurs périphériques.

Considérant cette théorie psychophysique des canaux, la perception des gros détails au détriment de détails plus fins s'imposerait au sujet du seul fait qu'il s'agit de la première facette de l'image source disponible pour les centres supérieurs. D'autres théories concurrentes, notamment dérivées des thèmes Gestaltistes, peuvent rendre compte de cette préférence globale sur l'analyse fine.

Ainsi, considérant ces deux approches (fonction de transfert de l'organe sensoriel et modèles de vitesse de transmission aux centres nerveux supérieurs) l'équipe progresse dans la construction d'un modèle des premiers stades de la perception applicable plus directement au concept d'aide intelligente.

1.2. Un certain modèle de la perception et les aides intelligentes en aéronautique : le projet *ERGO-IMAGE*.

Les connaissances actuelles ne permettent pas d'avoir l'ambition de dresser un modèle complet et fonctionnel de la perception visuelle partant d'un stimulus physique complexe et arrivant à une interprétation valide dans le contexte. Là n'est pas l'ambition de l'équipe ni du modèle.

Cette modélisation de certaines facettes des premiers stades de la perception est à beaucoup d'égards naïve car trop incomplète. Mais cette simplicité est d'un autre côté un gage de fonctionnalité pour des aides dont l'ambition est limitée. Voici le principe retenu pour les 2 systèmes d'études.

1.2.1. - le premier projet consiste - partant des caractéristiques physiques d'une source primaire et complexe - à en dériver une ou plusieurs images secondes correspondant à des étapes de filtrage et d'intégration que l'on peut supposer être - grâce au modèle - celle dont dispose le Système Nerveux Central.

L'aide se manifeste dans le système par cette capacité à reproduire un certain nombre d'étapes sensorielles afin d'orienter la perception vers telle ou telle facette de la réalité que, spontanément, le filtre perceptif avait masqué ou peu favorisé.

L'application choisie concerne les photo-interprétateurs d'images satellites, mais le principe d'une telle aide peut être appliqué aux visualisations du monde extérieur présentes dans les cockpits. Cette perspective est d'ailleurs envisagée à plus long terme.

Dans ce cadre, le programme de recherche débute actuellement sur 3 axes :

- Psychophysique : en recherchant par voie expérimentale quels filtrages sont évocateurs d'un contenu non évident en première lecture, le paradigme choisi est celui du test de GOTTSCHALDH sur les figures embrouillées. Ce test s'inscrit dans la problématique des styles cognitifs et plus particulièrement de la dépendance-indépendance à l'égard du champ (WITKIN,1978). Les sujets dépendants du champs ont du mal à analyser finement et analytiquement une image. On utilise ce constat pour déterminer différents filtrages des images sources afin de faciliter la perception des détails chez ces sujets. Ces filtrages pourraient par la suite être reconduits et testés sur des images complexes de photographies aériennes.

- Psychologique : en recherchant chez les professionnels de la photo-interprétation quelles règles de lecture ils acquièrent pendant leur formation (il s'agit ici de règles de lecture au sens exploration de l'image et non au sens de l'interprétation fine d'un objet donné).

- Informatique : en recherchant les algorithmes d'analyses en termes de contrastes spatiaux des images complexes et surtout en recherchant les différentes possibilités de filtrage, recomposition de l'image ainsi que l'intégration complète de règles de production destinées à conserver une *sémantique à l'image*.

1.2.2. - le second projet est plus directement appliqué à l'Aide Intelligente au pilotage : il s'agit de piloter les afficheurs afin qu'ils modifient, en parallèle aux capacités perceptives et dans les limites compatibles avec une conservation de l'information, les caractéristiques physiques des stimulations (en termes de fréquence spatiale, contrastes) lors des diverses agressions aéronautiques.

La faisabilité de ce projet nécessite l'obtention première de banques de données sur les modifications perceptives lors d'agressions aéronautiques (cette étape est en cours, les étapes de faisabilité technologique ne seront pas envisagées avant deux ans).

II - MODELES MENTAUX ET AIDES INTELLIGENTES

L'opérateur est par essence un être raisonnant et anticipant. Il dispose de connaissances que l'on peut appeler déclaratives ou cognitives sur son tableau de bord, sur sa mission. Il dispose aussi de connaissances plus dynamiques, plus fonctionnelles groupées de façon circonstancielles, que certains appeleront représentations mentales opératives (OCHANINE,1981) ou fonctionnelles (LEPLAT,1985) et d'autres modèles mentaux (NORMAN,1983).

L'ensemble de ces connaissances peut être lui-même désigné par le terme de "compétences sur le domaine" (de MONTMOLLIN,1983) ou encore par celui d'expertise (école de Carnegie Mellon, 1975).

Le rôle des modèles mentaux est de guider et de réguler les activités, d'indiquer ce qu'il y a à faire. On parle dans ce cas de connaissances procédurales. Sans entrer dans la célèbre controverse déclaratif/procédural (voir par exemple WINOGRAD, 1981), il est clair que l'on sait peu de choses sur les raisonnements réels de l'opérateur.

Les acquis à ce jour sont souvent centrés sur la résolution de problèmes ; les applications en tant qu'aides intelligentes interactives sont complètement tournées vers cette situation.

La situation de problème est ici classiquement définie (NEWELL & SIMON,1959) comme une situation pour laquelle existe un état initial, un état final à atteindre, des opérateurs de transformation d'état, et aucune solution connue pour passer d'un état à l'autre. Les modélisations informatiques de la résolution de problèmes simples, tel que le jeu de la "tour de hanoI", sont maintenant nombreuses. On citera la plus classique : General problem Solver de NEWELL et ALL. (1959) et quelques dérivés introduisant les capacités de généralisation et de raisonnement par analogie (CARBONELL, SAGE 2, UNDERSTAND...).

Ces modèles informatiques *ont connu de nombreuses tentatives* d'applications aux situations de contrôle de processus, notamment dans les industries nucléaires. Un courant de recherche très productif s'intéresse

ainsi aux situations incidentielles : résolution d'incidents graves (par exemple RASMUSSEN (1985) dans l'industrie nucléaire, SENACH (1986) dans le contrôle du trafic du métro...). Les méthodes employées pour recueillir la base de faits nécessaire au modèle sont soit l'observation sur site réel ou simulé (analyse de protocoles), soit des techniques de laboratoires (informations à la demande).

L'aide intelligente est souvent dans ce cas un système expert capable, en regroupant les évènements d'une certaine façon, de proposer des conclusions interprétatives sur la cause (en tout cas, au moins des orientations).

L'étude systématique des accidents aériens et les observations sur le terrain nous ont amené (AMALBERTI, 1986) à une réflexion un peu différente de ces travaux.

En effet un grand nombre d'accidents se produisent non parce que l'opérateur n'a pas sû résoudre une situation problématique mais parce qu'il n'a pas sû ou pas compris qu'il était dans une situation problématique : la première difficulté à résoudre une situation à problèmes c'est de savoir qu'il s'agit d'un problème.

Exemple : le boeing des Korean Air Lines s'écarte de plusieurs centaines de kilomètres de sa trajectoire sans que l'équipage perçoive cet écart ... le problème n'a pas été perçu.

Dans tous les cas, l'aide qui aurait été précieuse consisterait à forcer l'équipage à changer de représentation mentale, à reconsidérer la situation ou à ne pas focaliser son champ perceptif à quelques informations seulement.

C'est donc dans cette voie de développement d'aides intelligentes que le laboratoire s'est engagé.

Le principe du système proposé est le suivant :

Un module informatique travaille en parallèle au pilote ; il a pour charge une surveillance globale de la situation à court et moyen terme. Il évalue notamment les écarts entre la valeur instantannée de certains paramètres, la valeur de ces mêmes paramètres à moyen terme compte tenu du déroulement du vol, et une certaine valeur acceptable toujours de ces mêmes paramètres à court et moyen terme.

Son analyse est fondée sur le recueil et l'interprétation des actions du pilote sur l'interface, la connaissance des stratégies et des heuristiques les plus importantes nécessaires à la mission ainsi que la connaissance d'un déroulement formel de la même mission et de ses variantes les plus fréquentes.

Ce système devrait être capable à partir d'un calcul de cohérence de déceler les écarts et de les signaler au pilote. Pour être sûr que ce dernier percevra cette information d'alerte, ces informations peuvent être affichées en lieu et place d'autres informations que le module suppose, compte tenu de la stratégie détectée, que l'opérateur prélève régulièrement.

Ce type d'aide est complètement orienté vers la sécurité des vols à court et moyen terme. Il s'agit de minimiser les conséquences de certains inconvénients des comportements cognitifs humains (inattentions, routines, fixité contextuelle des représentations mentales ...).

La base du système interactif est la partie module de détection de contexte, module de détection de stratégies. Le projet "AIDE" développé au C.E.R.M.A. vise à établir la faisabilité d'un tel module sur un cas exemple, celui de la pénétration très basse altitude sans visibilité. Il intègre à plusieurs niveaux l'ergonomie cognitive et les systèmes informatiques intelligents.

Ce projet, conduit sur plusieurs années, comprend plusieurs étapes :

1 - mise à plat de l'expertise d'un pilote professionnel affecté à ce type de mission,

2 - modélisation informatique de cette expertise et généralisation à toutes les situations de la tâche et à tous les opérateurs.

3 - réalisation proprement dite du module intelligent pour ce cas exemple.

La phase 1 est réalisée, la phase 2 est en cours.

Les méthodes utilisées pour recueillir l'expertise sont basées sur une analyse de la tâche formelle, très fine, des techniques de questionnement de l'expert et l'observation sur le terrain de vols réels avec cet expert, AMALBERTI et VALOT (1985) ; AMALBERTI et al. (1986).

La formation s'articule autour de la notion de plans, schémas et scripts. Elle inclut aussi deux éléments plus originaux que sont "les règles d'univers", connaissances capables de moduler, inhiber ou favoriser des stratégies et les connaissances sur l'interface encore appelées "connaissances sémiologiques" qui sont les atomes élémentaires du savoir nécessaire à l'exécution des plans, schémas et scripts (figure 2).

Les plans correspondent aux différentes variantes globales de la mission étudiée.

Les schémas sont des groupements de connaissances permettant l'exécution de sous parties du plan. Pour chaque sous partie (e.g. navigation sur route préparée), l'expertise rend compte d' "un schéma prototypique" ou schéma de référence. Ce dernier est construit à partir de la synthèse des entretiens conduits avec le pilote expert. C'est un "exemplaire" qui n'a jamais été réellement observé mais qui représente au mieux les

actions de ce segment. D'autres variantes de ce schéma (incluant celles réellement observées sur le terrain) sont disponibles dans l'expertise. Elles permettent de moduler le schéma prototypique en fonction des contraintes extérieures.

Les scripts sont également des groupements de connaissances élémentaires destinés à l'exécution sur le système d'actions ou de séries d'actions décrites dans les schémas. Tous les scripts sont en un certain sens prototypiques : ils sont relativement rigides (leurs décours est peu influencé par les évènements) ; ils peuvent s'appliquer à une même situation et posséder le même but mais leur contenu reste très différent de l'un à l'autre. Le pilote en connait plusieurs et se sert de l'un ou de l'autre suivant le contexte. C'est pourquoi le choix des scripts est effectivement décrit dans l'expertise par un système de règles de production.

Les connaissances "sémiologiques" sont assimilables à des connaissances déclaratives. Elles font correspondre à un indicateur du tableau de bord une signification particulière avec des valeurs clés.

L'adaptation à tous les pilotes et toutes les situations des aides intelligentes les plus sophistiquées pose à propos de cette expertise les deux questions suivantes :

- peut-on compléter surtout pour les domaines de vol inhabituels (capacité à changer de stratégies, stratégies de sauvegarde). Les techniques d'entretiens et de simulation ne suffisent plus pour rechercher ces connaissances.

- est-elle stable ou quasi stable entre professionnels pilotes d'un même avion et d'expérience aéronautique comparable ?

Le système AIDE est destiné, dans un premier temps, à compléter l'expertise et à répondre à ces questions sur la généralisation. Il s'agit d'une modélisation informatique de l'expertise déjà possédée, capable d'intéractions temps réel avec un modèle informatique avion.

AIDE simule la conduite de processus du pilote. Il se caractérise par sa capacité à justifier sur son module sortie (écran cathodique) ses actions, les lieux où il prélève l'information (zoom sur la zone du tableau de bord consultée) et ses raisonnements (fenêtre d'explications en langage naturel et affichage des buts et sous buts poursuivis).

Le processus d'enrichissement et de généralisation repose sur la confrontation de ce fonctionnement "transparent" du modèle avec les pilotes de ce type d'avion. AIDE peut être interrompu à tout moment et dispose de capacités de playback. Son architecture permet également l'insertion aisée de nouveaux plans, schémas, scripts et autres connaissances.

2.1. Architecture et fonctionnement du modèle

2.1.1. Architecture

La structure informatique a été élaborée pour refléter au mieux "la structure cognitive" à laquelle aboutit l'expertise.

La mission est décomposée en plusieurs phases pratiquement autonomes. Chaque phase est un objet (au sens des langages orientés objet) avec une structure interne spécifique dont nous allons détailler un exemple (Cf. schéma).

La partie centrale est le moniteur communication qui centralise les flux d'informations et de décisions s'échangeant dans l'objet. Il joue le rôle d'un "blackboard". Le moniteur schéma contient la liste des scripts à accomplir par le module dans le cadre de l'objet ainsi que leurs spécifications. Les scripts proprement dits sont dans le dictionnaire de scripts.

Un moniteur "évènements", un moniteur "interruptions" et un moniteur "temps" sont reliés au moniteur "communication" et interviennent dans les mécanismes d'autoadaptation aux exigences de la situation.

Fonctionnement (figure 3) :

En fonctionnement normal, le moniteur "communications"donne la main au moniteur "schéma" initialisé avec un schéma prototypique. Ce dernier déroule sa suite de scripts et alimente la base de faits courants.

Dans le cas d'une réponse non attendue à un script, d'une panne, d'une interruption ou d'une alarme, la main revient au moniteur "communications" qui déclenche le moniteur "évènements" ou le moniteur "interruptions". Ces moniteurs disposent d'une base de règles spécifiques qui travaille sur la base de faits courants et décide du choix d'un ou plusieurs scripts de traitement de l'incident à insérer dans le moniteur schéma.

Le moniteur "temps" est ensuite appelé et à l'aide de règles également spécifiques reflétant le comportement du pilote face à la pression temporelle valide le nouveau schéma et décide des caractéristiques temporelles des scripts (durée, ordre de succession et même suppression de certains scripts).

AIDE est ainsi capable de s'autoadapter à un grand nombre de situations. Sa stratégie consiste à conserver la validité du schéma prototypique, aménagé plus ou moins fortement selon l'incident, aussi longtemps que possible. Il compilera les procédures, il sautera les étapes, il contrôlera moins de paramètres dans ce seul but de pouvoir effectuer ce schéma plus les corrections de l'incident dans le temps initialement imparti.

Ce mode de fonctionnement nous parait simuler correctement certains aspects du fonctionnement de la représentation mentale humaine. Cette stratégie a d'ailleurs quelques avantages puisqu'elle économise une révision complête de la situation (coûteuse en temps et en complexité) avec remise en cause du schéma voire du plan. AIDE est cependant capable, si aucune alternative ne s'offre à lui, de faire cette révision en situation incidentelle.

En relation avec ce programme purement opérateur tourne un modèle avion qui sert à simuler les évolutions de l'avion et une interface graphique pilotée par le modèle opérateur et qui présente l'état de la planche de bord de l'appareil. Selon les spécificités de l'objet et du script en cours, apparaissent des zooms sur la partie du moniteur concernée par les prises d'informations ou les actions. En parallèle défilent dans une fenêtre les justifications des stratégies et des scripts employés (figure 4).

2.2. Expérimentation

Le protocole de travail avec les pilotes prévoit deux types de scénarios à juger :

- les premiers sont des variantes du scénario prototypique de la mission dont on sait que AIDE est capable de résoudre en reproduisant des heuristiques observées chez l'expert.

- les seconds sont des variantes si contraintes que AIDE ne peut les résoudre par une simple application des heuristiques qu'il connait. Le modèle propose alors des solutions plus ou moins valides.

Chaque pilote prépare chaque scénario comme s'il allait exécuter la mission afin de disposer des connaissances factuelles de contexte et mieux juger de la qualité des solutions proposées par le modèle.

Il peut arrêter à tout moment le processus, revenir en arrière, verbaliser des corrections ou des écarts de façon de faire ; toutes les séances sont vidéoscopées et analysées secondairement afin d'incorporer au modèle les nouvelles connaissances ou stratégies.

A ce jour, AIDE est en cours de programmation. Il devrait être achevé en fin d'été 1987 et les expérimentations sont prévues à cette date. La construction d'un module d'analyse contextuelle applicable à cet exemple est envisagé pour 1989.

III - CONCLUSION

Depuis quelques années, l'avenir des recherches en ergonomie psychosensorielle passe par l'utilisation de systèmes informatiques intelligents. Nous avons essayé dans ce texte de montrer, à partir d'exemples développés au C.E.R.M.A., quelques pistes à cette évolution future des cockpits qui deviendront à la fois plus flexibles et plus personnalisés dans leurs présentations. Le plus grand bénéfice de ces systèmes sera finalement de combattre les défauts ou les limites psychosensorielles de l'opérateur tout en lui préservant ses qualités de décideur et d'acteur à tous les niveaux du contrôle de processus. Tout système participant à cette transformation mérite le nom d'ergonomique.

Une autre forme d'ergonomie consiste et consistera sans doute à automatiser, à décharger le pilote de tâches de plus en plus complexes. Cette voie réduit la charge de travail mais elle exclut aussi d'une certaine façon l'opérateur d'un plus ou moins grand nombre d'actions. Les études doivent se poursuivre avec assiduité pour la contrôler et l'appliquer à bon escient, c'est à dire justement de façon ergonomique. Ainsi, une meilleure connaissance des processus cognitifs du pilote permettra, même dans cette voie de l'automatisation, de faire fonctionner les machines avec une intelligence "plus humaine". L'opérateur pourra alors réellement comprendre son partenaire machine, le diriger et "reprendre la main" à bon escient.

BIBLIOGRAPHIE

AMALBERTI R.
Influence de la haute fiabilité du matériel aéronautique sur les prises de décisions du pilote.... Une forme d'effet boomerang.
Proceedings of the 5th Intern. Conf. on Reliability and Maintenability, 1986.

AMALBERTI R., VALOT C.
Task analysis in Aeronautics, a new approach.
Proceedings of the 2nd IFAC/IFIP/IFORS Conference. Johannsen G., Mancini G. & Martensson L. (Eds), 1985 : 380-385.

AMALBERTI R., VALOT C., MENU J.P.
Etude des mécanismes d'anticipation en Aéronautique.
AGARD C.P. 414, Information, management and decision making in advanced airbone weapon systems, 1986 : 1.1.-1.14.

BISSERET A.
Psychologie pour une co-opération H/M intelligente.
Actes Congrès IFIP, edit. AFCET, 1983 : 87-94.

BRADDICK O., CAMPBELL F., ATKINSON J.
Chanels in vision : basic aspects.
In : R. HELD, H. LEIBOWITZ, H. TEUBER (Eds), Handbook of sensory physiology, Vol. VIII, Berlin, Springer-Verlag, 1978.

CAMPBELL F., GREEN D.
Optical and retinal factors affecting visual resolution.
Journal of physiology, 1965, 181 : 576-593.

De MONTMOLLIN M.
L'intelligence de la tâche.
P. LANG Eds, 1985.

FITTS P., JONES R.
Analysis of factors contributing to 460 "pilot error" experiences in operating aircraft controls.
Report TSEAA-694 12, Air Material Command, Wright Patterson Air Force Base, 1947.

LEPLAT J.
Les représentations fonctionnelles dans le travail.
Psychologie Française, 1985, 30 (314) : 269-275.

MENU J.P.
Effects of colour on the contrast sensitivity function as a function of excentricity.
6th International Visual Field Symposium, Proceedings series, Vol. 42
Wunk Publisher, 1985 : 247-253.

MENU J.P.
THèse de Doctorat ès Sciences.
Université René Descartes, 1986 : 397p.

NEWELL A., SHAW J., SIMON H.
Report on a general problem solving program in Proceedings of the International Conference on Information processing, 1959, 256-264.

NORMAN D.
Some observations on mental models.
in Mental models, Stevens & Gentner Eds., Laurence Erlbaum Publishers, 1983, 7-14.

OCHANINE D.
L'image opérative.
Actes d'un séminaire et recueil d'articles.
Université de Paris I., 1981.

RASMUSSEN J.
Strategies for state identification and diagnosis.
in W.B. Rouse (eds) : Advances in Man Machine systems research, Vol. 1.
J.A.I. Press, 1985.

SENACH B.
La recherche de solution aux incidents en contrôle de processus.
Psychologie Française, 1984, 29 (3/4) : 279-283.

WIENER E.
Cockpit automation.
in Need of a philosophy.
SAE technical paper 851956, October 1985.

WINOGRAD
Frame representations and the declarative/Procedural controversy.
in Representation and understanding,Bobrow and Collins eds., Academic Press, 1975, 185-210.

WITKIN H.A.
Cognitive styles in personal and cultural adaptation.
Heinz Werner Lecture Series, 1977, vol. II.
Clark University Press, 1978.

Ce travail a été réalisé grâce au support financier de la D.R.E.T. (contrat 86.1054)

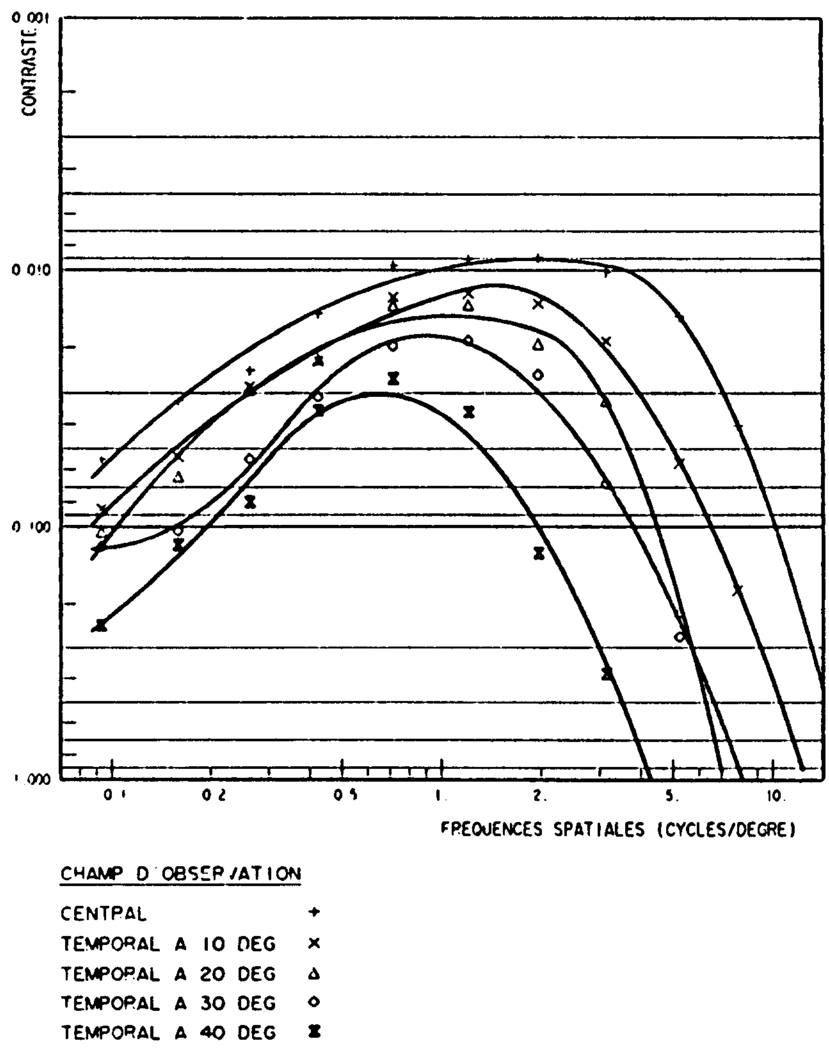


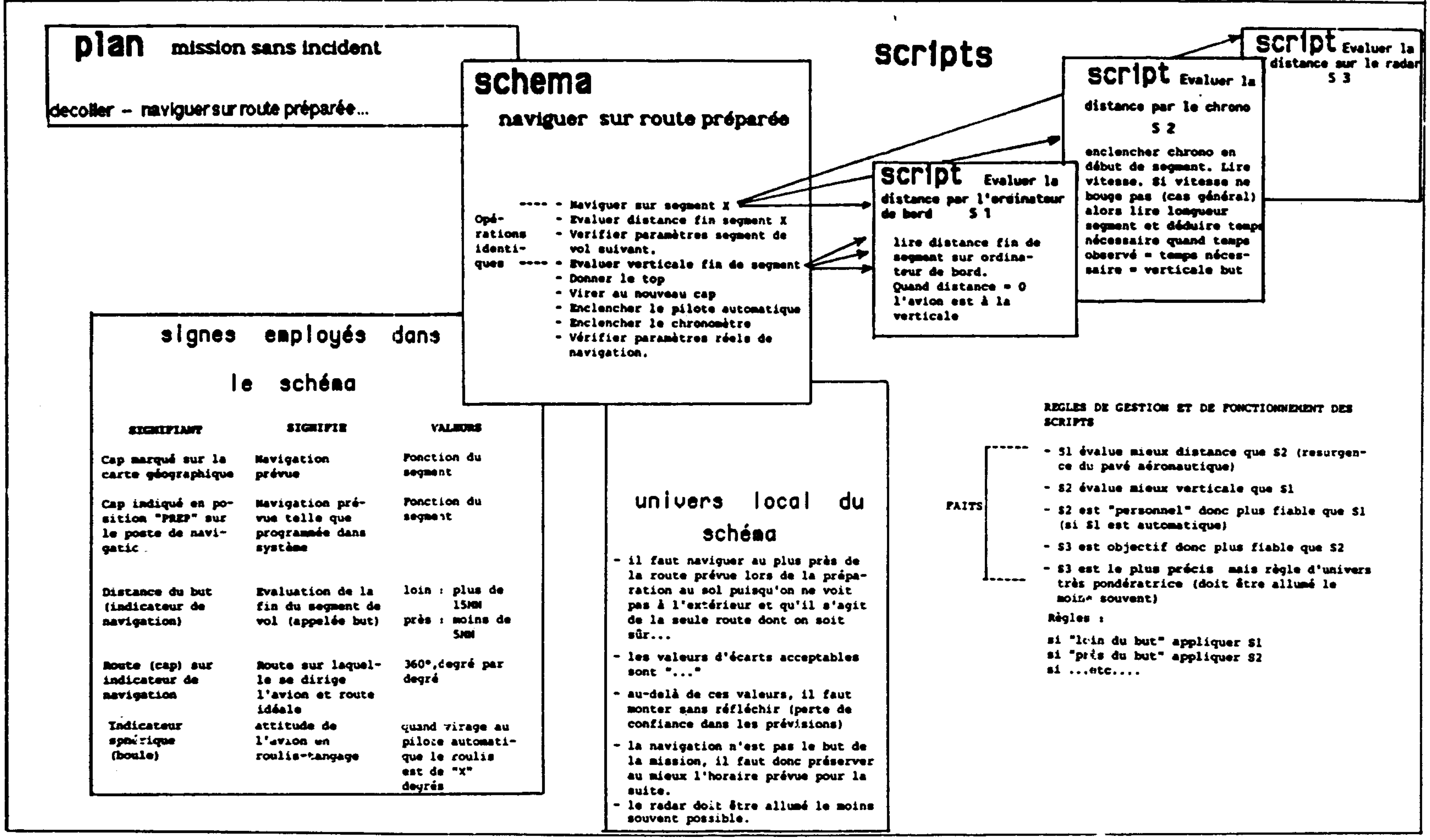
FIGURE 1

EVOLUTION DE LA FONCTION MOYENNE DE SENSIBILITE AUX CONTRASTES SPATIAUX POUR LA COULEUR BLEU (40 cd/m2) en FONCTION DE L'EXCENTRICITE DE PRESENTATION DE RESEAUX VERTICAUX

FIGURE 2

EXEMPLE DE MISE A PLAT DE L'EXPERTISE DU PILOTE DE COMBAT.

Cet exemple volontairement très simplifié illustre le rapport entre les différents niveaux de connaissances possédées par le pilote.



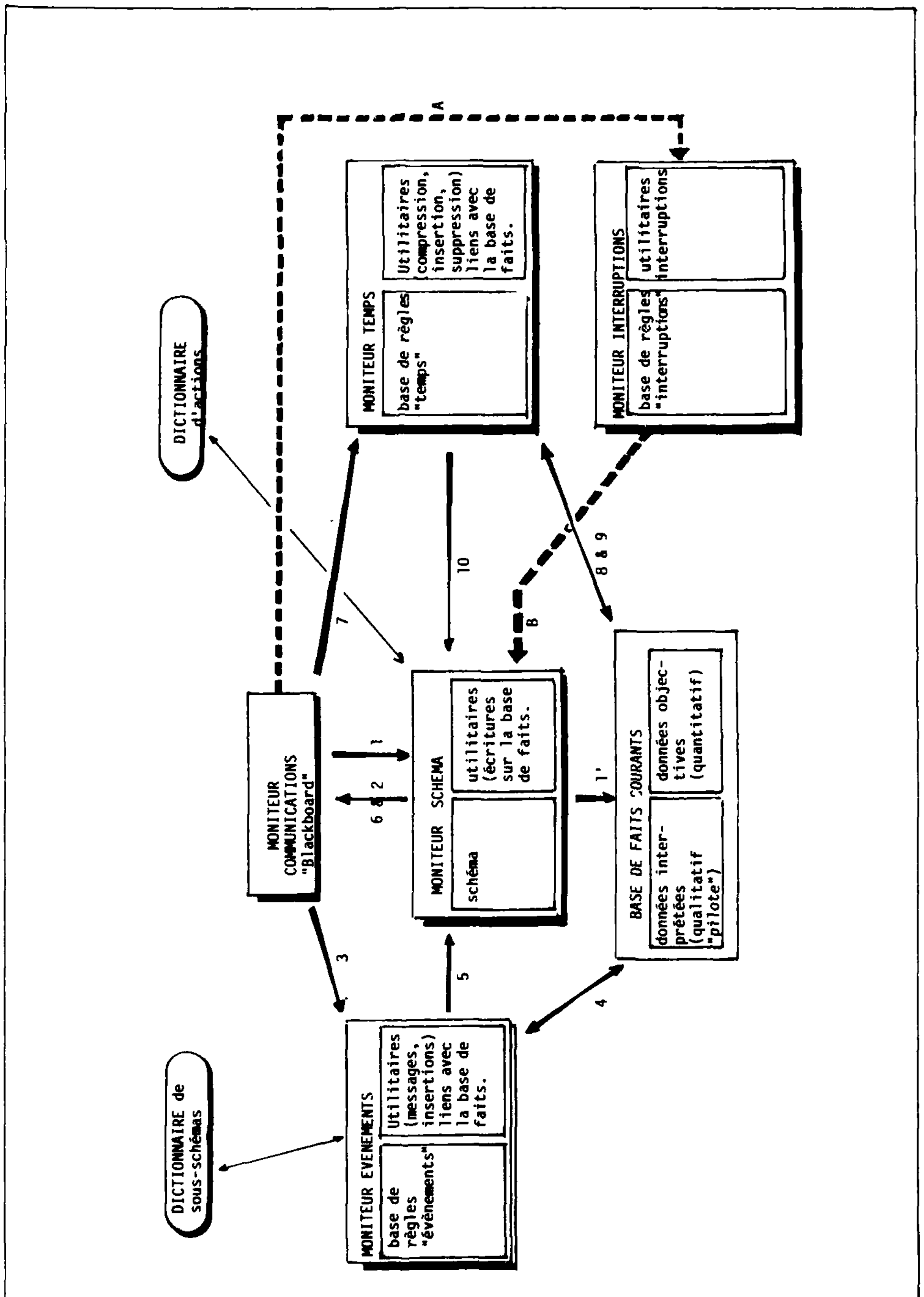
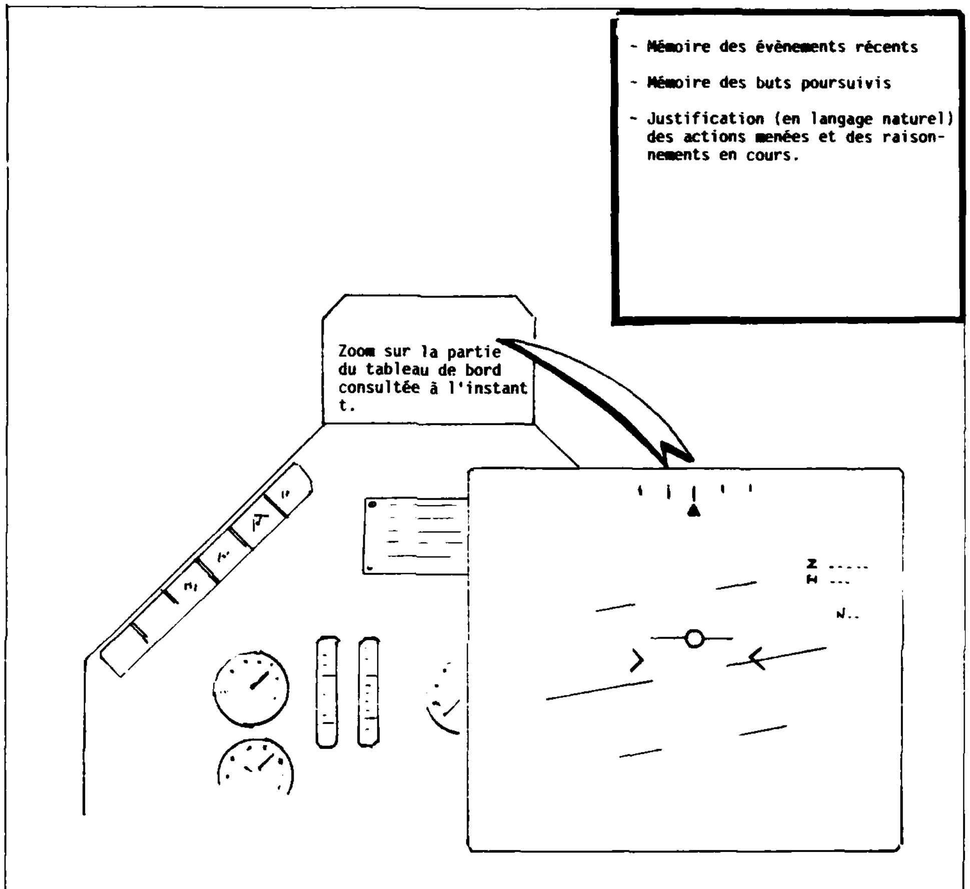


FIGURE 3

MODELE INFORMATIQUE : les différentes parties d'un objet avec leurs spécificités et leurs interactions

Les flèches 1 et 1' indiquent le déroulement d'un schéma sans évènement fortuit. Les flèches 2 à 10, A et B indiquent la manière dont est gérée l'adaptation du schéma aux contraintes extérieures.



<u>FIGURE 4</u>

SCHEMA DE SORTIE SUR ECRAN DU FONCTIONNEMENT DU MODELE INFORMATIQUE DE L'OPERATEUR.

Ce type de présentation est destiné à servir de support au recueil complémentaire d'expertise auprès des pilotes.

DISCUSSION

J.Nicol, UK

Does the research completed to date give the CERMA team confidence in the coherence of pilot mindset?

Author's Reply

Yes, we are confident in the relative stability of pilot mindset. On one hand, pilots obviously are human. They have opinions and, in separate conditions from real, they may vary their comments on the same situation, depending on the context and interlocutors.

On the other hand, increasing system complexity largely reduces the real variants of their own way to manage systems. New planes are more and more procedural in nature. Degrees of freedom are diminishing. For that reason, for the main heuristics at least, we are confident in the coherence of pilot mindset.

W.Mellano, IT

Which tools and languages did you use in formalizing the pilot expertise?

Author's Reply

We used VAX station GPX-11 (DEC) with live memory extended to 9 Moctets and LISP language in formalizing the pilot expertise.

ADVANCED DEVELOPMENT OF A COCKPIT AUTOMATION DESIGN SUPPORT SYSTEM

Philip V. Kulwicki
Joe W. McDaniel, PhD
Lisa M. Guadagna, 2Lt, USAF
AAMRL/HEX-CAT
Wright-Patterson AFB, Ohio 45433-6573
United States Air Force

SUMMARY

Under the auspices of the United States Air Force Cockpit Automation Technology (CAT) advanced development program, a highly disciplined and structured crew system design process (along with its supporting design tools and technology) is being developed to improve the efficiency and effectiveness by which advanced cockpits can be fielded. As an initial implementation of the CAT design process, a Cockpit Automation Design Support System (CADSS) is being developed to provide a computer-aided design environment, including the software design tools and the simulation utilities that can facilitate the development of the crew system in synchrony with the development of the avionics and weapon system. The rationale underlying the CADSS will be described in terms of the system components which include a Designer's Computer-Aided Design System (DCADS) processor, new software tools and a breadboard cockpit simulator which are envisioned to complement, but not replace, existing development facilities. This implementation of a cockpit design support system is described in relation to the overall CAT program activities and schedule.

INTRODUCTION AND BACKGROUND

The CAT program stems from a management recognition of the increasing importance of the crew system within the overall avionics and weapon system design, the lack of a standard process for development in the current design practice, and the need for a systematic design process having traceability and an audit feature to help avert (but, if necessary, to correct) design flaws early in the weapon system acquisition cycle. The CAT design process is tightly coupled to the overall weapon system development and concentrates on satisfying operational mission demands and aircrew needs in an iterative manner, with a formal connectivity among engineering analysis, design and piloted simulation.

Origins for the CAT project date to 1980, at which time the United States Air Force Systems Command (AFSC) planned for a new crew systems technology initiative to focus on the needs of combat crews. Prior to that time, those needs were addressed in other programs, but were sometimes underemphasized due to the hardware and software complexities of emerging systems. Under the leadership of the Aerospace Medical Division (recently renamed the Human Systems Division), a component of AFSC, three advanced development projects were established (see Figure 1). An Advanced Life Support Systems Project was activated in 1983 to provide a new generation of aircrew personal protective equipment (to include new flight gear for altitude and acceleration protection and for other purposes). In 1984, an Advanced Crew Escape System Technology Project began to advance the state-of-the-art for technologies needed to expand the escape envelope associated with open ejection seats. Also initiated in 1984, the Cockpit Automation Technology (CAT) Project was established to organize and advance the process for crew system design (see Figure 2).

NEED FOR DESIGN PROCESS DEVELOPMENT

The CAT project is unique in that it concentrates on process rather than aircraft components or point designs. The need for investment in design process could be questioned, in view of today's scarce research and development resources, because (after all) we have designed and fielded operational systems having substantial capability. However, the current development practice has led to a situation where: (1) important design deficiencies are identified very late in development (this leads to excessive reliance on expensive changes in both hardware and software), (2) the weapon system development itself is a lengthy process (often spanning more than 10 years), (3) the cost and complexity necessary for advanced weapon systems are continuing management concerns, (4) the systems remain in operational inventory for decades and undergo significant configuration upgrades, and (5) significant crew system operability and workload concerns are becoming apparent in emerging systems.

In the arena of cockpit crew systems, the emergence of digital avionics, with its potentially massive amount of data available for display and large number of aircrew

control interactions, raises new development challenges. Today's aircrew members, particularly in fighter systems, must monitor and control a myriad of sophisticated avionics systems and weapons, often in hazardous flight regimes, with demanding time constraints, in concert with cooperative aircraft participating in the same mission, and under potential threat attack. Pilot workload is sometimes excessive, partly due to the need for sorting through encyclopaedic quantities of data (much of which, in current systems, may be irrelevant to the task at hand) to glean that which is necessary to perform the immediate task. The consequence, in the modern combat arena, can be task "shedding" (i.e., some needed tasks are not performed) or loss of situation awareness which, once lost, may be difficult or impossible to regain. This occurs, in the authors' opinion, because of serious limitations in the crew system design process (as it is currently applied; see Figure 3).

Based on three years of detailed examination of both development needs and current practices in crew system design/evaluation, we observe that:

(1) There is little or no standardization of the current crew system design process. There are company-to-company differences in design process and its application; there are system-to-system differences in the extent of management oversight.

(2) There is an over-reliance on expert opinion in crew system design decisions, with minimal dependence on formal analyses and truly objective tests.

(3) Often, steps are "skipped" and shortcuts are taken in order to meet schedule.

(4) There is an over-reliance on time-consuming manual design methods. In this regard, the crew system design community appears to seriously lag other technical disciplines in the field of design automation.

(5) Crew system design tends to be emphasized late in the system acquisition process with other related design constraints already fixed; this inhibits the trade-off of crew system needs with other subsystem needs.

(6) Crew system design guide documents are outdated. For example, according to Smith & Mosier (1986, pp4-5), "human engineering standards and design handbooks have in the past been of little use to the software designer ". Often, design decisions are not recorded and, if recorded, they are not easily retrievable. Crew system design "lessons learned" are not effectively distributed to new system developments.

(7) While there has been a substantial increase in the field of engineering flight simulation, crew system developers have been relatively slow to exploit its potential. Objective measures to evaluate crew system performance are argumentative (or missing). Simulator tests may not employ scientific controls and they often focus on subjective preferences for proof-of-concept rather than on engineering data to support design trades.

(8) Often, high fidelity flight simulators are not directly accessible to crew system developers; tools that are readily accessible, such as static mock-ups and simplified part-task design aids, tend to be useful more for form-fit-function studies than for understanding pilot information management and workload as they relate to the crew system operation (yet this latter design area is critical for mission suceas).

(9) Mission requirements established for air vehicle sizing and performance prediction are insufficiently detailed for crew system development.

(10) The direct interrelationship between crew system development activities and other related weapon system development activities appears to be poorly understood, leading to an inefficient generation and use of design data; the extent and focus of crew system development varies as the design progresses toward deployment of the weapon system.

(11) During the system lifetime, a substantial amount of design change is inevitable, but a scheme to maintain the traceability of the crew station design evolution has not been implemented. As a result, change decisions can be based upon unnecessarily incomplete information which can aggravate the reintegration problem.

(12) Although attempts have been made to develop computer-based design "tools", the developments have been somewhat piecemeal and the resulting products are not generally in wide use.

OVERVIEW OF COCKPIT AUTOMATION TECHNOLOGY PROJECT

The CAT project has been planned and organized to redress the above concerns. Use of the term "automation" in the project title confers the expectation that automation technology will be central to resolving the difficulties that must be resolved in fielding effective combat crew systems. This project develops an organized, methodological crew system design and development process; it applies this process to a specific operational mission to demonstrate proof-of-concept; it develops a computer-based support system as a design environment to promote design process

efficiency. Collectively, these developments can help to elevate the crew system to greater status in the weapon system work breakdown hierarchy. Demonstrated gains in crew system development effectiveness and efficiency are equal goals. Based on clearly stated (projected) operational requirements, the CAT process will progress through stages of mission analysis, functional analysis/allocation, integration and design, and progressive test/evaluation. This auditable design process will determine the need for interrelating weapon system automation with crew system operation based on mission demands, aircrew abilities and technology cost/benefit. The extent of automation will be established, based upon formal engineering trade analyses, real-time piloted simulation tests, and objective aircrew performance and workload measures. The CAT process, in particular, confronts the information needs of combat aircrews to achieve mission success with realistic, manageable aircrew situation awareness and workload. As such, the crew system can become a recognized aircraft subsystem (of equal importance to airframe or propulsion in the weapon system development).

The CAT project delivers several related products: a systematic and detailed crew system design process, a cockpit automation design guide, proposed revisions to existing design guides and standards, a cockpit design specification derived by applying the CAT process to a specific fighter mission, a breadboard computer-aided design/engineering system as an initial implementation of the CAT process and its software support tools, prototype instrumentation for effective measurement of pilot performance and workload in ground and airborne environments, and verification of the design process using scientifically controlled, engineering flight, full mission simulations. This paper concentrates on the computer-aided support system, which we term the Cockpit Automation Design Support System, or CADSS (see Figure 4). The CADSS integrates new design tools and techniques into a self-contained support system for the crew system design team.

Programmatically, the CAT project is organized in a number of technical phases. Phase 1 was a definition phase and provided an exploration of cockpit design process improvements. Phase 2 is currently underway and involves the full development of the CAT design process and the partial development of the CADSS. Phase 2 has two industry teams in direct competition (one or both may be continued through Phase 3, the demonstration phase, which is expected to commence in May 1988). For this reason, specific details of the respective design approaches for the CADSS are not presented herein; the discussion below is applicable to both Phase 2 technical approaches. Lastly, a fourth technical phase is planned and will further demonstrate and disseminate the resulting crew system development process, and its implementation in hardware and software, to industry and Government organizations. We call this phase a technology transition phase.

CADSS CONCEPT FORMATION

The idea of using computer information tools to support cockpit analysis, design and evaluation is not new. Despite some noteworthy tools, in general, crew system development remains predominantly a manual process. This is partly because cockpit development has been by trial-and-error engineering which (in the past) did not require such support. Another likely reason is that the cockpit design necessarily involves human factors, which are difficult to characterize with computer analysis. For example, researchers still disagree on fundamental theories of human behavior for routine situations (not to mention performance under combat stress). This is particularly the case for the aircrew's mental processes as they relate to task performance, decision-making and workload. In early days of aircraft development, mission requirements and weapon systems were less complex and the aircrew could adapt so as to make up for cockpit design shortcomings. Today, however, the aircrew can no longer be regarded as a "slack variable" and, thus, over-reliance on a purely experimental engineering approach to crew system development is insufficient.

Researchers in the past two decades have anticipated that today's crew system design process must be supported by modern computer tools. According to Lind (1986): "Previous attempts to do this have yielded products of low general utility and have failed, in general, to take into account the overall vagueness, subjective quality, and limited integrability of much of the subject material. Moreover, the individual programs support very limited portions of the design process and make limited provision for the manner in which the crew systems designer actually does his or her work and for the scope of the work to be done. The crew systems designer, and in fact any designer, usually does not work in a linear fashion, designing one step completely before moving on to the next; rather, the designer will lay out a rough design which is then refined through many iterations. No integrated computer environment to support this style of work exists."

Lack of success of many early attempts may have contributed to a piecemeal development of computerized design tools. Such efforts concerned isolated, stand-alone, parts of the problem. Difficulties have been noted with excessive reliance on user expertise, with awkward and labor intensive set-up of input data, with complex and voluminous data output which requires lengthy interpretation (in large system developments design questions must be resolved quickly), with incompatible operating systems and programming languages (which prevent independently developed tools from working together), and with other difficulties.

The current generation of computer hardware and software technology has significant advantages over the systems with which previous design tools were configured. Across-the-board advances are underway in speed, memory, software and even artificial-intelligence hardware architectures. These can now be employed in the attempt to provide a practical computer support environment for crew system design in service of the CAT process. Part of this recent progress is associated with the rapid development and introduction of engineering workstations, data base management systems, and computer-aided design/engineering (CAD/CAE) software. The result is a greatly increased graphics and file manipulation capability, which appears necessary to implement a practical CADSS.

The CADSS must support the practical needs of the crew system designer. It is unrealistic to expect that purely analytic tools would suffice, just as it is unrealistic to expect a great advance in design technology would accrue from totally relying on the experimental engineering legacy from the past. We envision that the CAT design process must employ both analysis tools and experimental tools and, for that reason, have provided both capabilities in our CADSS concept.

The CADSS concept is illustrated in Figure 5. In a very "top level" description, this system is comprised of four components.

(1) DCADS. The most critical element of the CADSS is what we term a Designers Computer-Aided Design System which hosts the main data handling portion of the CADSS. It contains the major computing hardware and operating software of the CADSS. Though depicted as a single block in the figure, it may be separated into an array of supporting elements. The purpose of the DCADS is to allow the cockpit designer to use computerized data bases, computer models, computer analysis tools, simulations, and computer drafting tools. The system will have a general purpose central processing unit as its core and, as shown in the diagram, the DCADS will provide a simulation executive for real-time, piloted, part-task tests with a breadboard simulator discussed below. Also included in the DCADS are several engineering workstations. These workstations represent a major input/output environment for the designer. Many of the CAE and analysis tools needed for crew system development will be hosted at the workstations, but each workstation will readily communicate with the DCADS central processor.

(2) CAT Software Tools. These tools are being configured specifically to support the designer in using the formal crew system design process being fully developed in the CAT project. The CAT software tools are in two categories: commercially available and custom developed/adapted. Because they are mature and have been developed outside the CAT project, some commercially available software tools will be incorporated into the CADSS. For example, we envision that the crew system designer will need ready access to data bases of various natures. Accordingly, a commercially available data base management system will be needed. Likewise, a variety of computer-aided-design software packages are available. Commercial software may be incorporated to the extent that it is needed for CAT, mature, documented, and likely to be supported and maintained. Other software will be custom developed or adapted for use in the CADSS. Specifically falling in this category are three analysis tools and a lessons learned data base which are being devised in Phase 2 of the CAT project. The analysis tools will support the crew system designer in mission decomposition, function analysis/allocation, and information analysis. Associated with these analysis tools will be one or more on-line computer data bases, also to be developed in the CAT project. Such state-of-the-art tools are not commercially available. Additional tools in this category are also shown in Figure 6.

(3) Breadboard Cockpit Simulator. The crew system designer needs convenient access to a real-time simulation device with which to quickly test new design ideas. Mock-ups used for general layout and installation (to check clearances, for example) are not adequate to use for design decisions concerning many critical issues including display content/format, control procedures and pilot workload. Due to a relatively high cost as well as a demand from other design disciplines, high fidelity dome simulators are often not directly accessible to the crew system designer. When a design question arises it usually needs immediate resolution and often does not require the full mission dome capability. In the CADSS, we make provision for a very flexible, real-time, part task simulator that is easily modified (both hardware and software) to quickly test crew system design concepts. Importantly, this device is under the direct control of the crew system design team and is customized for the CAT process.

(4) Software Executer System. Some users of the CAT design process, such as Government managers, may require only a design/analysis support capability. This system will allow those users to execute the CAT software tools without requiring direct access to the DCADS or to the breadboard cockpit simulator. Because of the relief from driving a real-time, man-in-the-loop simulation, it will be possible to configure this system on a smaller and less expensive computer than used for the DCADS. The system is envisioned to operate in a typical office environment and will be easily transportable. In particular, this system will include all CAT software tools pertinent to the evaluation of cockpit design concepts.

CADSS DEVELOPMENT AND DEMONSTRATION

Using the above formulation of the CADSS, two independent industry teams are

preparing designs which will be developed in hardware and software. Figure 7 shows which parts of this effort are underway in the development phase (CAT Phase 2) and which will be undertaken in the demonstration phase (CAT Phase 3) and in the transition phase. Detailed requirements for the CADSS are being independently derived from the CAT design process being developed by each team. Where possible, this section will describe common requirements and design considerations.

The relationship of the CADSS to the CAT crew system design process is shown in Fig 8. Here, four specific stages to the CAT process are illustrated. Both the CAT process methodology and the application of the CAT process are shown in schematic. Note that the CADSS is employed in each stage of crew system development, from the initial conception and requirements definition through analysis/design/integration and including test and evaluation. Two important points, not illustrated, should also be noted.

First, crew system development is a highly iterative process which does not lend itself to this kind of diagram. In order to have a common basis for evaluating the two CAT processes under development, we have imposed that the CAT process be described in IDEF notation (see Figure 9). IDEF stands for ICAM (Integrated Computer Aided Manufacturing) Definition and is a means of organizing process information according to well-defined rules. Figure 10 shows the arrangement of IDEF notation in terms of activity, inputs, outputs, constraints (controls) and resources (mechanisms). For the CAT application, the terms "constraints" and "resources" better illustrate the intended meaning, while the terms "control" and "mechanism" were part of the original IDEF formulation. The advantage of this notation is that it permits a progressive hierarchical decomposition of process activity into its lowest constituent elements (see Figure 11). Each granular block of process activity can then be examined in terms of potential CADSS usage. To illustrate this point, Figure 12 shows an IDEF representation for the entire weapon system. The crew station portion highlighted therein can be inspected in its own IDEF format, as in Figure 13, which details the top level inputs, outputs, constraints and resources. As envisioned during CAT Phase 1 (Quinn, 1986), this top level activity is then exploded into constituent activities in the manner shown in Figure 14. Each constituent activity itself might be further decomposed in the same manner. Importantly, all of the inputs, outputs, constraints and resources of the "parent" diagram would be reflected in the appropriate "child" diagram.

Again referring to Figure 8, the second important point to be made is that this diagram depicts crew system design relationships in the CAT process. Yet, in application to the overall weapon system development, these relationships will be affected by the evolving maturity of the weapon system itself. That is, the design emphasis, activity and labor level of effort (and hence, CADSS usage) will change depending on the phase of the weapon system acquisition process. In the United States these weapon system process phases are known as Concept Exploration, Demonstration and Validation, Full Scale Engineering Development, and Production and Deployment. Crew system design activity changes with passage from weapon system phase to weapon system phase. The CAT process is being developed in tight coupling with the weapon system process (see Figure 15). Therefore, CADSS design requirements must consider not only the hierarchical granularity of necessary crew system activity noted above, but also which weapon system development phase is currently active. These two important dimensions to setting the CADSS development specifications are significant. The CADSS, and the CAT process it supports, will be judged by its demonstrated improvements in design efficiency and design effectiveness.

Demonstrating the value of the CADSS in terms of design efficiency and design effectiveness are separate problems. True validation in these areas is infeasible because: (1) there is no accepted, well-understood standard of comparison for the "current" crew system process and its supporting methods and tools, and (2) the scope of the CAT project is confined to ground-based simulation for proof-of-concept. Subjective estimates of design efficiency will be possible by applying the CAT process (itself generalizable beyond a specific pre-defined operational mission) to a specific Government furnished mission scenario (requirement). Using the CAT process, supported by the envisioned CADSS support environment, a specific cockpit crew system is being designed in Phase 2 and will be fabricated and tested in full-mission simulation in Phase 3. It will be possible, through objective measures of task effectivenes in the Phase 3 real-time simulation, to infer design effectiveness. This will be approached by direct comparison of the CAT crew system against a baseline crew system derived by "conventional" crew system design practice (using common missions, weapons, threats and pilots for this comparison). These demonstration events have yet to be planned in detail. However, there is provision during Phase 3 for 600 hours of real-time, piloted, full mission combat simulation (exclusive of testing with the breadboard cockpit simulator described above).

The technical approach being followed in both ongoing CADSS developments is toward an integrated system that will support all crew system analysis, design and evaluation activities throughout all weapon system development phases. Although well over 100 existing computer software programs have been evaluated for potential use in the CADSS, only a relatively small number of key tools will effectively aid in crew system development. Considerable differences exist in the Phase 2 approaches for CADSS architecture (both hardware and software) and envisioned CADSS application. At this

time, a comparison of approaches is premature. In general, the CADSS is being designed to:

(1) Support the CAT analysis software algorithms,

(2) Support the CAT data bases and data base management system,

(3) Support the CAD packages adopted for the CAT process,

(4) Support the incorporation of new design evaluation tools as may be developed outside the CAT project,

(5) Support the CAT crew system evaluation tools,

(6) Support the real-time breadboard cockpit simulation,

(7) Support downloading the appropriate DCADS software to the Software Executer System envisioned for use in an office environment,

(8) Support industry standard interfaces to permit the broadest dissemination of CAT products,

(9) Permit an upwardly compatible path for system growth,

(10) Support the configuration management and design decision traceability of crew system development.

TECHNOLOGY TRANSITION CONCEPTS

Particular attention has been directed toward the eventual uses of the CAT crew system design process and its supporting CADSS. In the final analysis, system process and process support tools will be accepted and used only when the practical advantages are understood and demonstrated. Transition of this kind of technology requires more than a letter of transmittal and receipt of equipment. The CAT project is being developed and managed by an interdisciplinary, inter-organizational USAF team with representation of the Human Systems Division, the Harry G. Armstrong Aerospace Medical Research Laboratory, the Avionics and Flight Dynamics Laboratories of the Air Force Wright Aeronautical Laboratories, and the Aeronautical Systems Division, Deputy for Engineering (ASD/EN).

Responding to the above concern, the CAT project office has established a formal Technology Transition Plan which represents an agreement naming ASD/EN as recipient of the technology. This organization, in turn, will adapt the technology for exploitation in major system development. Specifically identified in this transition plan is a separate, funded transition effort (noted above) which will assist ASD/EN in testing, acquiring, using and supporting the deliverables. It is expected that the CADSS described in this paper will be upgraded by a second generation variant in that transition development.

Also related to the Transition Plan, the CAT project office has consolidated an understanding with the Advanced Tactical Fighter (ATF) System Program Office. This agreement provides for making CAT project developments available for use in that program. Accordingly, the CAT project schedule is coordinated with, and technical progress is monitored by, the crew system engineering function in the ATF development. This close coordination assures ASD participation in CAT project planning, decisions and milestones. More importantly, this presents an opportunity to test the viability of the developing crew system design process and the CADSS in a realistic development setting and to make necessary adjustments. The active involvement of the engineering development "customer" in an advanced technology development enterprise is considered noteworthy with respect to planning for the technology transition.

CONCLUSIONS

Current design methodologies available for crew system designers are embodied in design handbooks (AFSC DH 1-3 and 2-2), military standards (MIL-STD-1472C and MIL-STD-1776) and military specifications (MIL-H-46855). These methodologies are too general, antiquated in terms of the technology assumed, and do not answer the paramount question: "what does the crew need?" These methodologies focus on general cockpit layout, control/display arrangements, and anthropometric studies that support cockpit installation and physical fit. They also provide for part-task simulation which helps define cockpit procedures and function. Up until the mid-1970s, this procedure was adequate due to limited alternatives in cockpit componentry automation and avionics, a more limited flight envelope and a (comparatively) benign threat. The situation since then has changed in that we now have the capability to completely overwhelm the aircrew with complex control switchology, uninterpretable displayed data, and automation modes which may or may not help. Crew system design methods have not kept pace with air-vehicle, avionics and weapons advances.

The USAF Cockpit Automation Technology Project attempts to correct this situation

by state-of-the-art advancement in crew system design technology. The focus of this work is with a computer-supported, methodological design process that seeks to impose discipline on today's practice (with all its current limitations). A critical part of the CAT project is the Cockpit Automation Design Support Subsystem, without which it would be unrealistic to expect significant gains in either crew system design efficiency or design effectiveness. Some of the envisioned users of the CADSS are depicted in Figure 16. Because of the interaction of the crew system with other components of the air vehicle (Figure 17), the CADSS may eventually be used in conjunction with other computer based support systems that will likely evolve.

The CAT project is advanced technology development. For the first time, a critical mass of funding, along with inter-organizational cooperation within both Government and industry, is being brought to bear on a state-of-the-art advance in the cockpit design practice. Even partial success of this ambitious project will contribute to future gains, both in developmental efficiency and in operational suitability. Greater success can lead to reductions in development time and acquisition costs (in minimizing the need for engineering changes with their attendant inefficiencies). Together with the advanced life support and escape system projects, the Cockpit Automation Technology Project offers a man-centered orientation for future system development.

REFERENCES

1. Aldrich, J. R. et al, Phase 1 Final Technical Report for the Cockpit Automation Technology Project, Contract F33615-84-C-0517, Volume 1, "Description of the Refined CAT Design Process," The BDM Corporation, Dayton, OH, March 1986.

2. Aretz, A. J., "Cockpit Automation Technology", in Proceedings of the 1984 Human Factors Society Annual Meeting, San Antonio, TX, 22-26 Oct 1984.

3. Davidson, J. (Chairman), Report of the Committee on Automation in Combat Aircraft, National Academy Press, Washington DC, 1982.

4. Eggleston, R. G. and P. V. Kulwicki, "A Technology Forecasting and Assessent Method for Evaluating System Utility and Operator Workload," in Proceedings of the 1984 Human Factors Society Annual Meeting, San Antonio, TX 22-26 Oct 1984.

5. Eggleston, R. G. and P. V. Kulwicki, "Estimating the Value of Emerging Fighter/Attack System Technologies," in Proceedings of the NATO Defense Research Group Panel VIII Workshop on Application of Systems Ergonomics to Weapon System Development, Shrivenham, England, April 1984.

6. "ICAM Architecture, Part II, Vol IV, Function Modelling Manual (IDEF)", Technical Report AFWAL-TR-81-4023, Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB OH, 1981

7. Kidwell, G. H., "Workstations Take Over Conceptual Design," Aerospace America, pp 18-20, January 1987

8. Kulwicki, P. V., "Addressing Human Factor Options in Conceptual Design," in AGARD Conference Proceedings CPP-266 on Operational Roles, Aircrew Systems and Human Factors in Future High Performance Aircraft, Lisbon, Portugal 22-26 October 1979.

9. Kuperman, G. G. and P. V. Kulwicki, "Mission Scenarios for Cockpit Automation Technology," in Proceedings of the AIAA/IEEE Sixth Digital Avionics Systems Conference, Baltimore MD, 3-6 December 1984.

10. Lind, H. O., "A Network Management Tool Applicable to Crew Systems Analysis and Design Techniques," Document No. D180-29158-1, Boeing Military Airplane Company, 26 November 1986.

11. Littlefield, J. E. and R. L. Howard, "Computer Graphics Win Over Engineers," Aerospace America, pg 22, January 1987.

12. McDaniel, J.W., "Cockpit Automation Technology: A Process for Designing Advanced Aircraft Systems", in Contemporary Ergonomics, Durham, England, pp 143-147, April 1986.

13. McNeese, M. D., Rik Warren, B. K. Woodson, "Cockpit Automation Technology: A Further Look," in Proceedings of the 1985 Human Factors Society Annual Meeting, Baltimore MD, 29 Sep - 30 Oct 1985.

14. Mitchell, A. R., "Market Supremacy Through Engineering Automation," Aerospace America, pp 24-27, January 1987.

15. Morgan, D. R., T. A. Furness, A. J. Aretz, D. E. Cole, and P. V. Kulwicki, "Cockpit Automation Technology Concepts," in Proceedings of the NATO Defense Research Group Panel VIII Workshop on Application of Systems Ergonomics to Weapon System Development, Shrivenham, England, Apr 1984.

16. Pew, R. W., et al, Phase 1 Final Technical Report for the Cockpit Automation Technology Project, Contract F33615-84-C-0516, BBN Laboratories Incorporated, Cambridge MA, June 1986.

17. Quinn, T. J., et al, Phase 1 Final Technical Report for the Cockpit Automation Technology Project, Contract F33615-84-C-0500, McDonnell Douglas Corporation, St Louis MO, May 1986.

18. Smith, S.L. and J.N. Mosier, "Guidelines for Designing User Interface Software, "Technical Report ESD-TR-86-278, Electronic Syserms Division, AFSC, Hanscom AFB MA, August 1986.

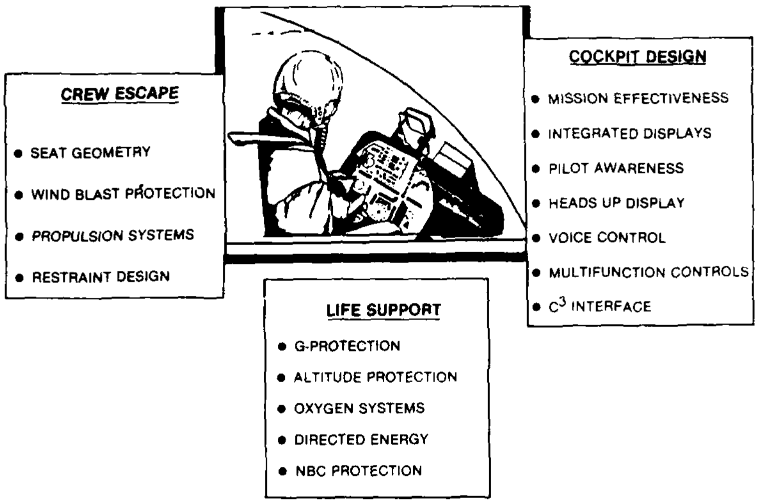


FIGURE 1. CREW SYSTEMS INTEGRATION.

DEVELOP AND DEMONSTRATE A CREW SYSTEM DESIGN PROCESS

- **DEVELOP DESIGN METHODS, TOOLS AND TECHNOLOGY**
- **APPLY TO FIGHTER WEAPON SYSTEM**
- **DESIGN TO { COMBAT REQUIREMENT / PILOT CAPABILITY**
- **DETERMINE BEST USES FOR COCKPIT AUTOMATION**

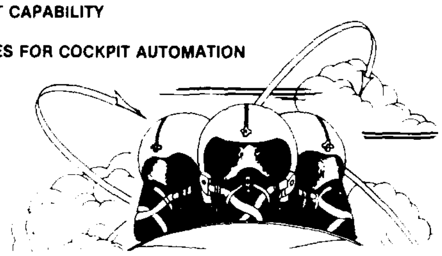


FIGURE 2. CAT PROJECT OBJECTIVE.

- NOT STANDARDIZED
- TOTALLY PEOPLE DEPENDENT
- SCHEDULE DRIVEN
- MANUAL PROCEDURES
- APPLIED LATE
- DESIGN GUIDES OUTDATED
- SIMULATION UNDEREXPLOITED
- SIMULATORS INACCESSIBLE
- SCENARIO DETAIL LACKING
- WEAPON SYSTEM PROCESS RELATION
- MANAGING COCKPIT CHANGES
- MINIMAL COMPUTER TOOLS

FIGURE 3. LIMITATIONS OF CURRENT PROCESS

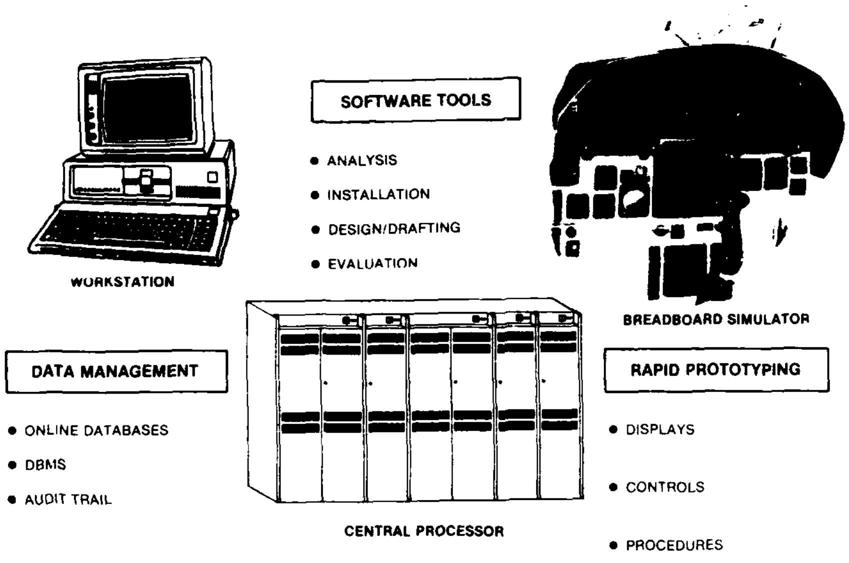


FIGURE 4. CADSS — CONFRONTING THE LIMITATIONS.

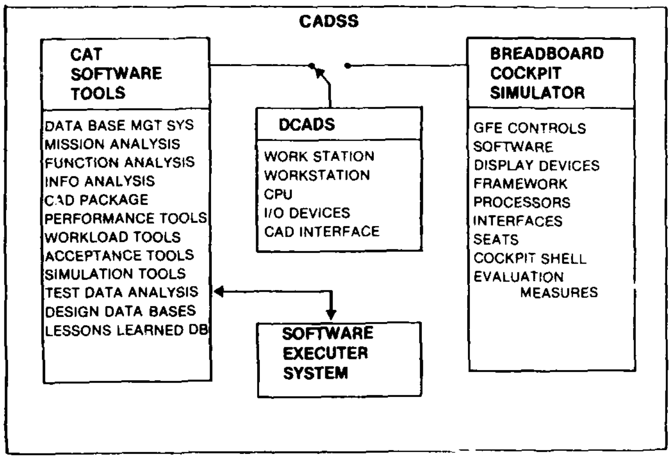


FIGURE 5. COCKPIT AUTOMATION DESIGN SUPPORT SYSTEM (CADSS).

- MISSION DECOMPOSITION TOOL
- FUNCTION ANALYSIS/ALLOCATION TOOL
- INFORMATION ANALYSIS TOOL
- DATA BASE MANAGEMENT TOOL
- COMPUTER AIDED DESIGN TOOL
- SIMULATION SOFTWARE TOOL
- PILOT PERFORMANCE EVALUATION TOOL
- WORKLOAD EVALUATION TOOL
- PILOT ACCEPTANCE TOOL

FIGURE 6. CAT SOFTWARE TOOLS.

	COCKPIT AUTOMATION TECHNOLOGY PROJECT PHASES		
CADSS COMPONENTS	**PHASE 2**	**PHASE 3**	**TRANSITION**
OCADS	FULLY DEVELOP AND DEMONSTRATE	EMPLOY AND UPGRADE	SECOND GENERATION
CAT SOFTWARE TOOLS	DBMS CAD PACKAGE MISSION ANALYSIS FUNCTION ANALYSIS INFO ANALYSIS LESSONS LEARNED	PERFORMANCE EVAL WORKLOAD EVAL PILOT ACCEPTANCE DATA ACQUISITION SIMULATION S/W	MODIFY FOR SES ADD REFORMATTER
BREADBOARD COCKPIT SIMULATOR	DESIGN STUDY PREPARE SPECS	FULLY DEVELOP AND DEMONSTRATE	UPGRADE
SOFTWARE EXECUTER SYSTEM	DESIGN STUDY PREPARE SPECS	COORDINATE REQUIREMENTS	FULLY DEVELOP AND DOCUMENT

FIGURE 7. CADSS EVOLUTION.

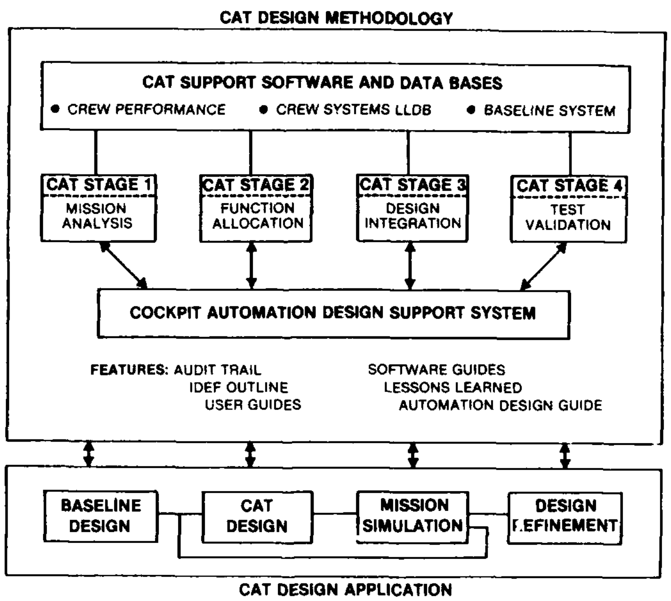


FIGURE 8. RELATION OF CADSS TO CAT METHODOLOGY.

- DISCIPLINED GRAPHIC CHARTING TECHNIQUE EMPLOYING RIGID RULES AND CONVENTIONS FOR ORGANIZATION AND REPRESENTATION OF **PROCESSES**
- CHARACTERIZE HIERARCHICAL AND CONNECTIVE RELATIONSHIPS AMONG PROCESS ELEMENTS, INPUTS, OUTPUTS, RESOURCES AND CONSTRAINTS
- PROVIDES BACKUP GLOSSARIES AND NARRATIVE DESCRIPTIONS OF GRAPHICS

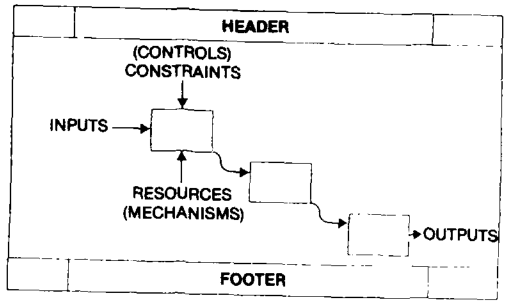


FIGURE 9. IDEF$_0$ FEATURES.

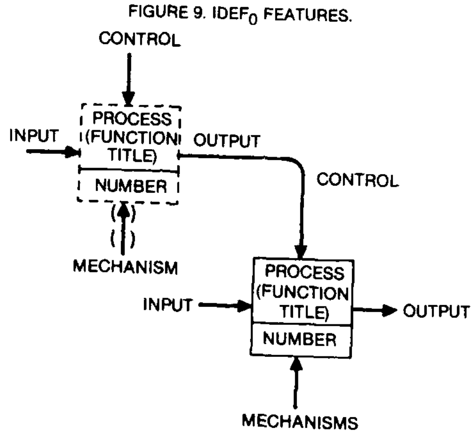


FIGURE 10. TYPICAL IDEF$_0$ FORMAT.

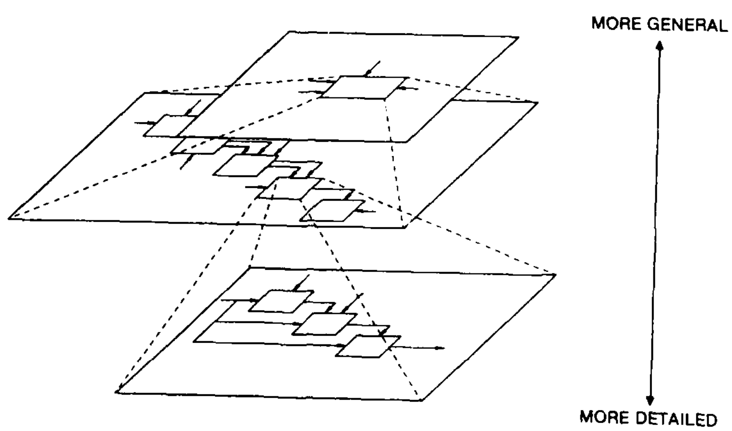


FIGURE 11. IDEF$_0$ PARENT-CHILD RELATIONSHIP.

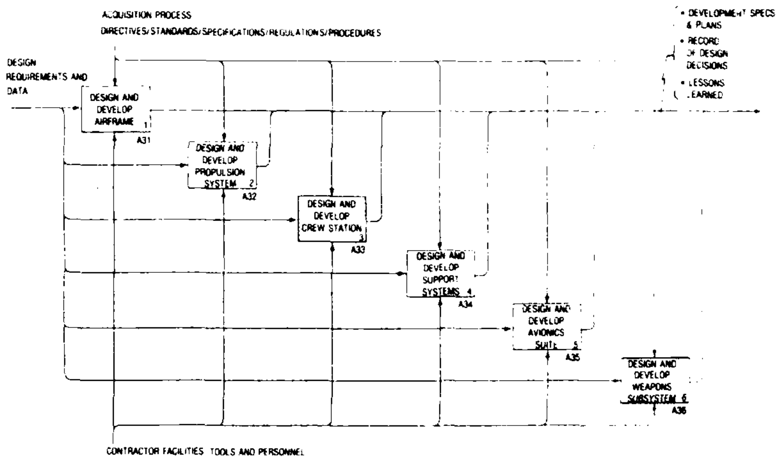


FIGURE 12. IDEF FOR WEAPON SYSTEM DEVELOPMENT.

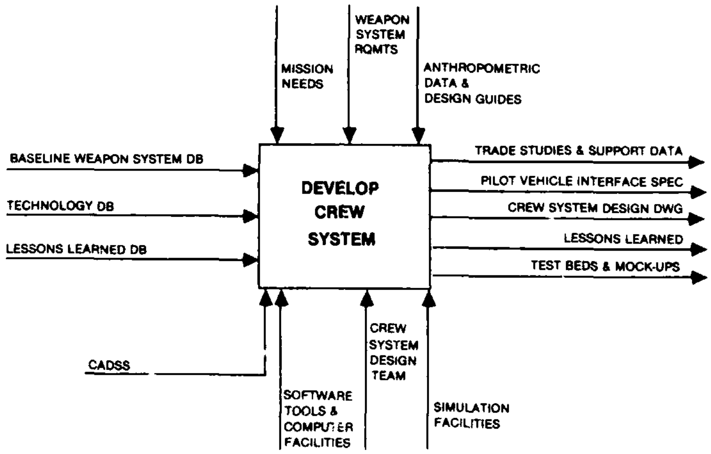


FIGURE 13. IDEF FOR CAT DESIGN PROCESS.

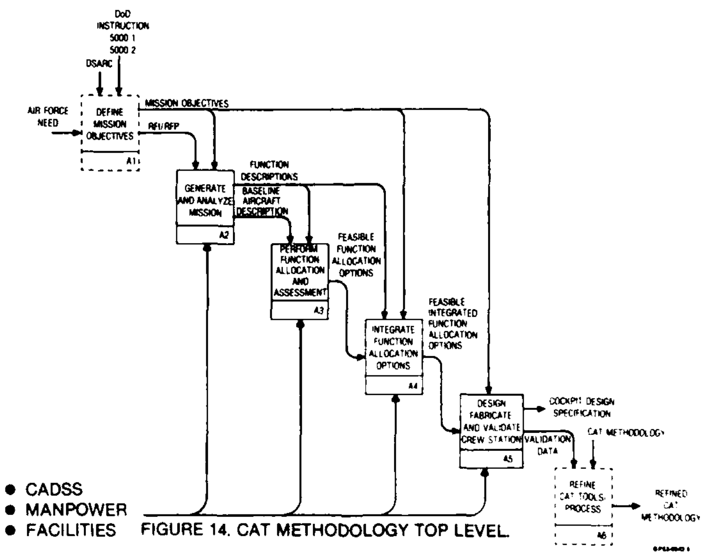


FIGURE 14. CAT METHODOLOGY TOP LEVEL.

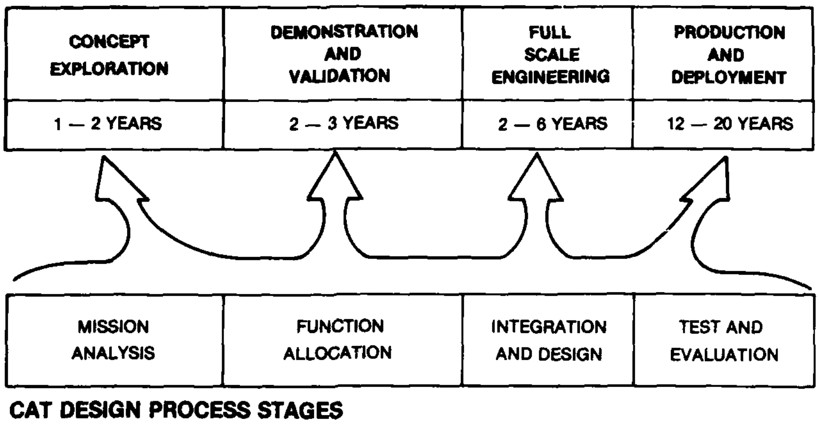


FIGURE 15. RELATING CAT PROCESS TO WEAPON SYSTEM PROCESS.

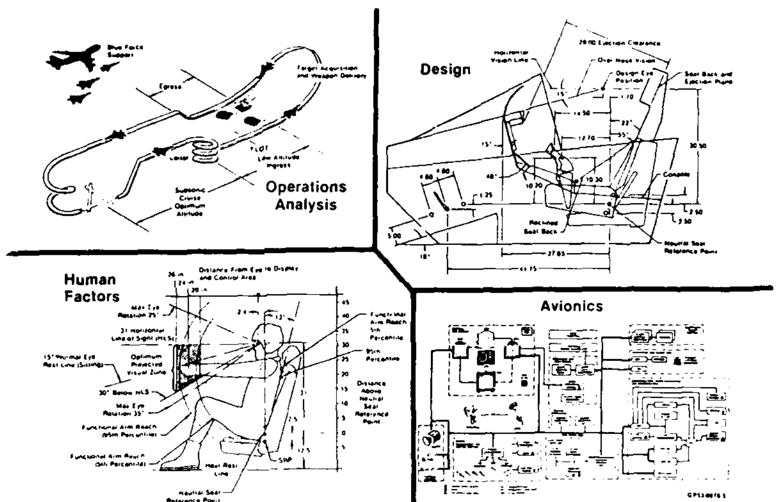


FIGURE 16. PRIMARY CADSS USERS.

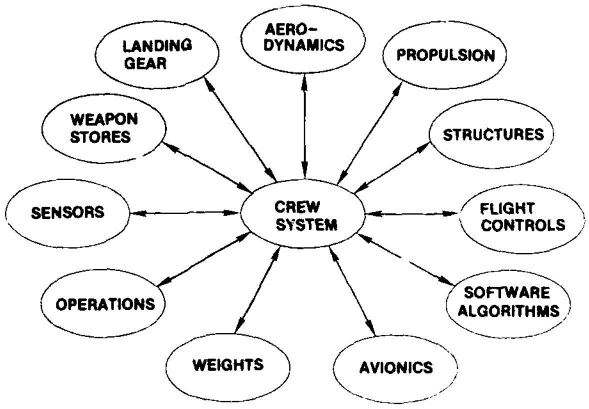


FIGURE 17. INTERDISCIPLINE INTERFACES.

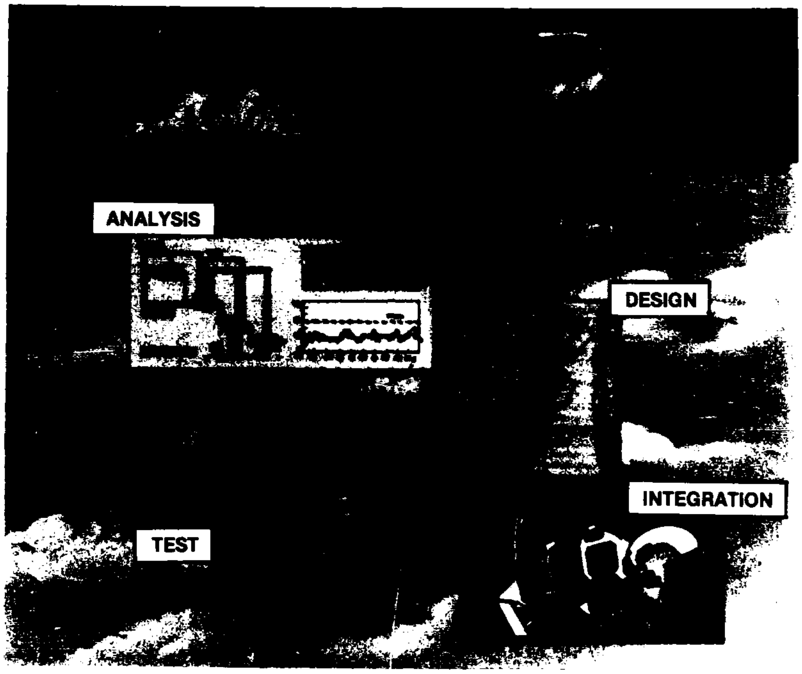


FIGURE 18. COCKPIT AUTOMATION TECHNOLOGY DEVELOPMENT
AND DEMONSTRATION.

CONCEPTION ET DEVELOPPEMENT D'UN SYSTEME AVIONIQUE ADAPTE AUX MISSIONS DES HELICOPTERES

par

Daniel BOUHERET _ AEROSPATIALE
Division Hélicoptères
B.P. 13
13722 MARIGNANE - FRANCE

et

Jean Louis ROCH _ CROUZET S.A.
Division Aérospatial
25 rue Jules Védrines
26027 VALENCE CEDEX - FRANCE

RESUME

Cette conférence présente l'organisation et les moyens mis en place pour le développement du système avionique des DAUPHIN 365 F S.A.R de l'AEROSPATIALE, et en particulier du sous-système de navigation et de gestion de mission développé par CROUZET. Après une description des missions de l'avion et de l'architecture du système, l'accent est mis sur l'organisation des travaux, entre avionneur et sous-systemier, puis sur les différents moyens mis en oeuvre chez l'équipementier comme chez l'avionneur pour assurer le succès d'un tel développement.

INTRODUCTION

La conception et le développement de systèmes avioniques complexes font appel à de nombreuses techniques et mettent en oeuvre plusieurs intervenants et différents moyens.

Ainsi l'avionneur, qui assure lui-même en tant que maître d'oeuvre du système la conception globale des spécifications du système, puis la responsabilité des phases d'intégration au sol et d'essais en vol, se trouve confronté à la gestion d'un projet complexe, dont les intervenants principaux sont constitués par les équipementiers majeurs qui fournissent les éléments essentiels du système.

Une méthodologie rigoureuse est nécessaire pour mener à bien ce projet, elle inclut notamment le respect d'une grande rigueur au niveau des spécifications, une étroite coopération entre l'avionneur et les fournisseurs, ainsi que l'utilisation de moyens cohérents et complémentaires.

Un exemple significatif est constitué par le système avionique spécifiquement développé par l'Aérospatiale en collaboration avec différents équipementiers (CROUZET - SFIM - BENDIX) pour l'équipement des hélicoptères de recherche et de sauvetage en mer. Ce système a été notamment mis en oeuvre sur les DAUPHIN 365 F livrés à l'Irish Air Corps.

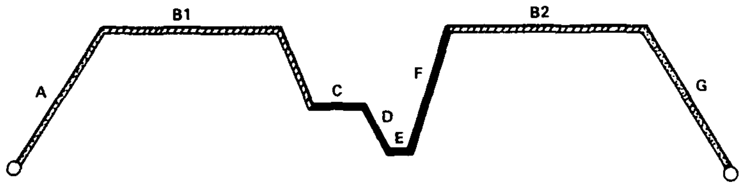
Les relations avionneur-équipementier sont décrites à travers la coopération entre l'Aérospatiale et la société CROUZET fournisseur du système de navigation.

1. PRESENTATION GENERALE DU SYSTEME

1.1 DESCRIPTION DE LA MISSION

La mission principale de l'hélicoptère est la recherche et le sauvetage en mer.

Cette mission peut être décrite à partir d'un schéma représentant le profil de vol :



Ce profil se décompose en huit (8) phases :

- phase A : *décollage+montée à l'altitude de croisière*
- phase B_1 : rejointe de la zone de recherche
- phase C : trajectoires de recherche (Pattern de recherche)
- phase D : descente vers le stationnaire (Transition down)
- phase E : stationnaire (HOVER). Opérations de treuillage
- phase F : montée vers l'altitude de retour (Transition UP)
- phase B_2 : croisière retour
- phase G : descente - approche - atterrissage

Les phases A - B_1 - B_2 - G sont des phases classiques du vol. Les phases C - D - E - F correspondent aux phases de la mission SAR proprement dite.

phase C – L'hélicoptère est arrivé sur la zone de recherche. Dans cette phase il devra couvrir cette zone jusqu'à localisation de son objectif.

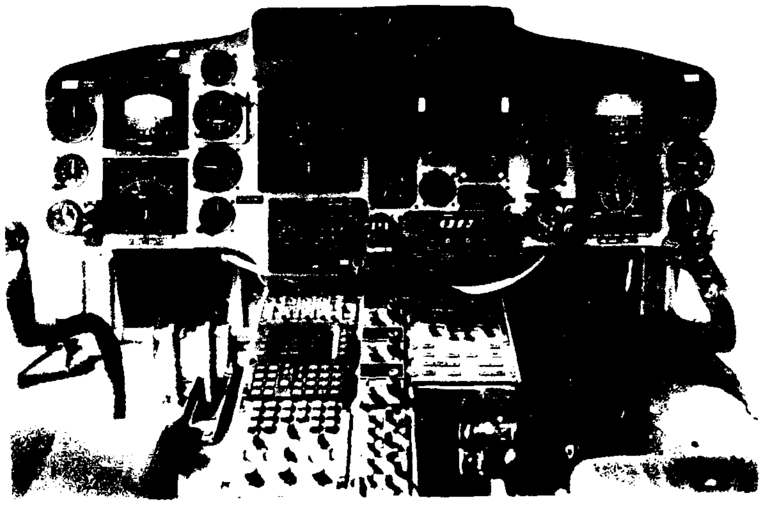
phase D – Cette phase consiste à effectuer une transition vers le bas pour venir se placer en stationnaire au-dessus de l'objectif.

phase E – Maintien du stationnaire jusqu'à la fin de l'opération de treuillage.

phase F – Transition vers le haut pour rejoindre l'altitude de croisière retour.

1.2 ARCHITECTURE DU SYSTEME

Pour assumer cette mission, l'hélicoptère choisi est un dauphin naval de la classe 4000 kg équipé d'un système de mission incluant des visualisations de planche de bord à écrans cathodiques.



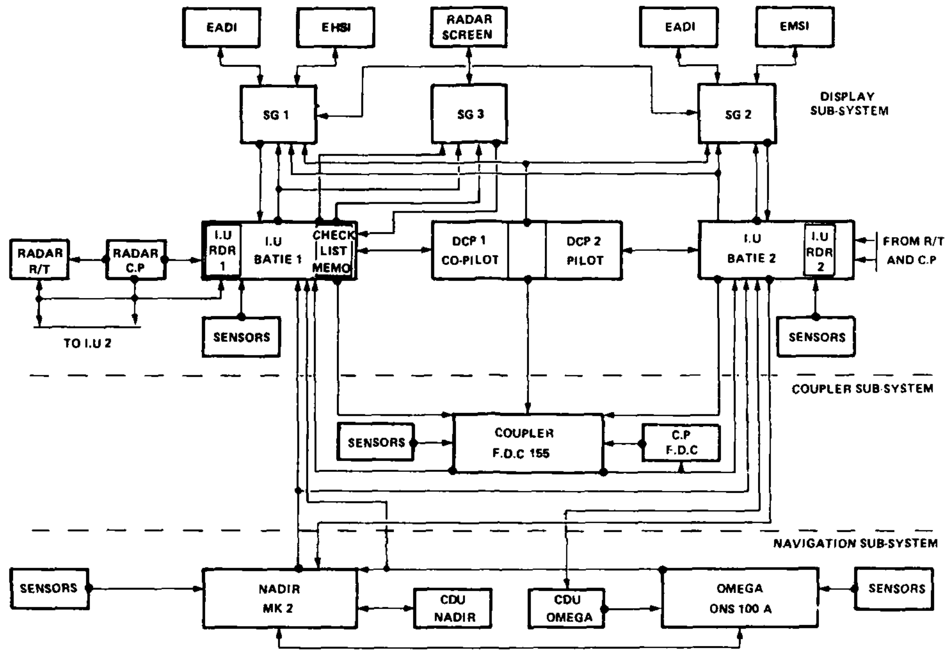
Le système de mission est scindé en trois sous-systèmes :

- un sous-système navigation et de gestion de mission

- un sous-système pilotage automatique

- un sous-système radar et visualisation

Ce système est parfaitement intégré et utilise des liaisons numériques pour les dialogues principaux. Son architecture est schématisée par le diagramme suivant :



EADI :	*Electronic Attitude Display Indicator*
EHSI :	*Electronic Horizontal Situation Indicator*
S.G :	*Symbol Generator*
B.A.T.I.E :	*Boîtier d'Adaptation et de Traitement d'Informations Electriques*
I.U :	*Interface Unit*
D.C.P :	*Display Control Panel*
F.D.C :	*Flight Director Coupler*
C.D.U	*Control and Display Unit*
O.N.S :	*OMEGA Navigation System*

Cette architecture est organisée autour de 3 calculateurs :

- *Le calculateur coupleur directeur de vol qui assure les fonctions classiques de pilotage automatique du vol et la mise en stationnaire automatique*

- *Le calculateur récepteur de navigation OMEGA*

- *Le calculateur NADIR MK2*

La suite de l'exposé décrit plus particulièrement le sous-système de navigation et de gestion de la mission

1.3 LE SOUS-SYSTEME DE NAVIGATION ET DE GESTION DE MISSION

Le sous-système de navigation et de gestion de mission est articulé autour de deux calculateurs. Un calculateur principal qui assure la totalité de la gestion navigation et un récepteur OMEGA qui en fonctionnement normal a un rôle de senseur de position.

En cas de défaillance du calculateur principal, le récepteur OMEGA retrouve sa fonction calculateur de navigation plus récepteur OMEGA et assure automatiquement la poursuite de la navigation en cours.

Le senseur OMEGA est un EQUINOX ONS 100 A de la société CROUZET.

1.3.1 Calculateur principal NADIR MK2

Le calculateur principal de navigation est le NADIR MK2 produit également par CROUZET.

Ce calculateur a été conçu pour satisfaire les exigences des systèmes modernes actuels et futurs, notamment en ce qui concerne la puissance de calcul, la taille mémoire, et la capacité d'entrées-sorties.
Ainsi, dans l'exemple du DAUPHIN 365 F, le calculateur NADIR MK2 est relié :

- *Aux différents senseurs lui permettant d'assurer une localisation multi-senseurs :*
  - *. Cap (compas gyromagnétique)*
  - *. Attitude (gyroscope de verticale)*
  - *. Radar Doppler de navigation*
  - *. VOR/DME (2 VOR - 1 DME)*
  - *. OMEGA*
  - *. Boîtier Capteur de Pressions permettant le calcul des paramètres anémobarométriques*

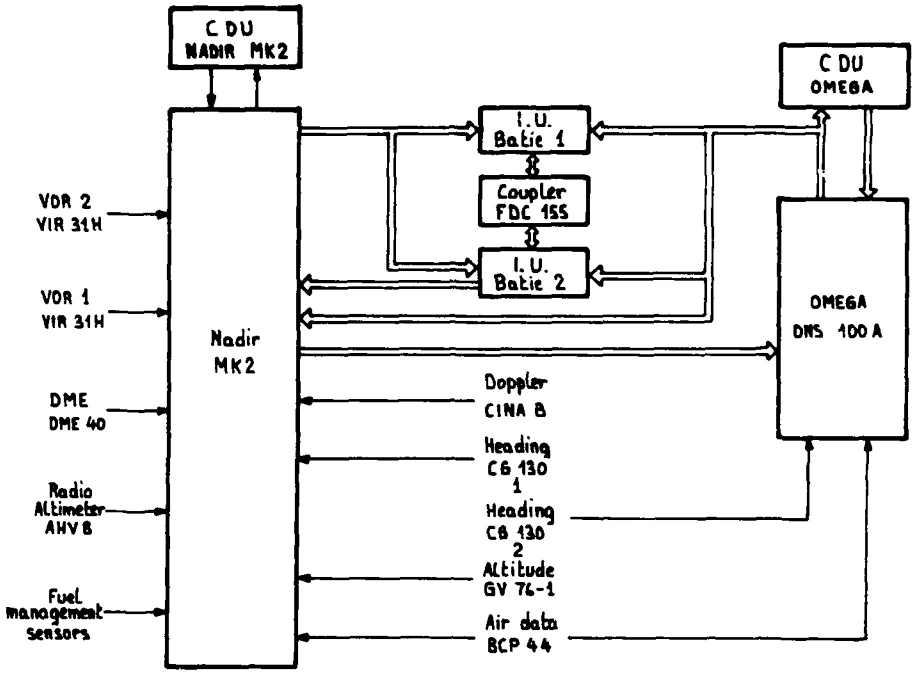
- *Aux débitmètres de carburant, ce qui permet d'entretenir en permanence les paramètres consommation, masses machine et carburant, distance franchissable, et d'assurer les fonctions de gestion carburant, gestion des performances.*

- *Au pilote automatique, au radar, et aux instruments de pilotage électroniques, à travers les boîtiers d'interfaces "BATIE" 1 et 2.*

- *Au radio-altimètre.*

- *A son poste de commande et de visualisation (PCV à tube cathodique).*

Le synoptique du sous-système est décrit ci-après :



Les fonctions principales du calculateur de navigation NADIR MK2 sont résumées ci-après :

- *NAVIGATION :*

  Trois positions présentes sont élaborées en permanence, à partir des informations :

  - *. Doppler*
  - *. OMEGA*
  - *. VOR/DME*

- *GESTION DU VOL*

  - *. Mémorisation des buts et des routes*
  - *. Gestion du plan de vol et des patterns de recherche*
  - *. Calcul des paramètres de navigation sur la route*

- *GESTION CARBURANT*

  - *. Consommation, masses carburant et machine*
  - *. Vitesse de croisière économique*
  - *. Calculs de masse décollable*
  - *. Calculs prédictifs de distance franchissable*

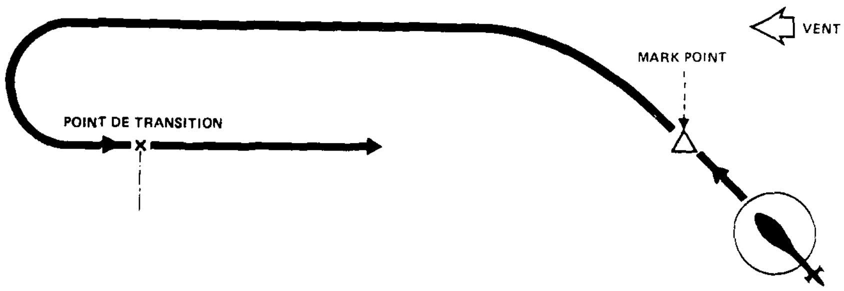
- *COUPLAGE PILOTE AUTOMATIQUE ET INSTRUMENTS*

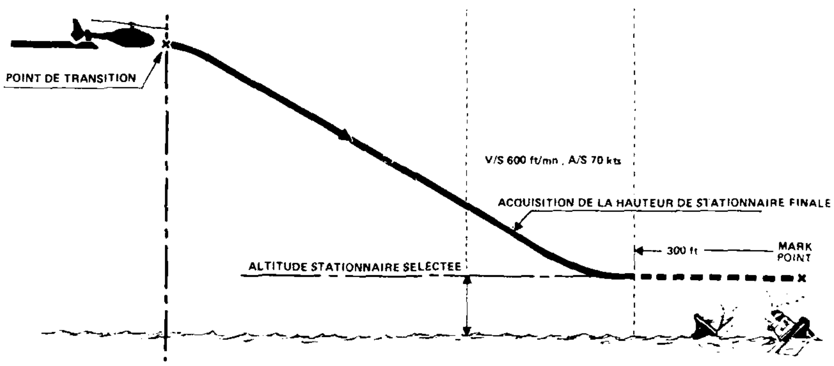
- *SURVEILLANCE ET RECONFIGURATION DU SOUS-SYSTEME*

Six modes de guidage sont possibles :

- *FROM/TO*
- *DIRECT TO, avec ou sans radiale preferentielle*
- *Ralliement d'un but mobile*
- *Navigation sur une route*
- *PATTERNS de recherche*
- *Mise en stationnaire (HOVER) apres transition automatique*

EVOLUTIONS GUIDEES EN HORIZONTAL ET VERTICAL LORS DE LA MISE EN STATIONNAIRE AUTOMATIQUE





1.3.2 Processeur utilisé

Pour réaliser l'ensemble de ces fonctions, une importante puissance de calcul est nécessaire. C'est pourquoi, dans la conception du NADIR MK2, CROUZET a utilisé une Unité Arithmétique très puissante, l'ALPHA 732, entièrement conçue pour les besoins avioniques.

Il s'agit d'un processeur 32 bits, travaillant en virgule fixe ou virgule flottante, de la classe de 1 million d'opérations par seconde.

Ce calculateur est programmable en PASCAL.

Un atelier logiciel très puissant a été conçu par CROUZET pour permettre une production aisée de logiciels importants ainsi que pour assumer une grande capacité d'évolutions et de modifications, ces caractéristiques étant essentielles pour la réussite d'un programme d'intégration de système complexe comme celui du DAUPHIN SAR 365 F.

Ainsi, tant dans la conception matérielle que logicielle du calculateur ALPHA 732, ont été prises en compte les contraintes liées à l'intégration dans un système avionique important :

- fonctionnement multi-tâches,

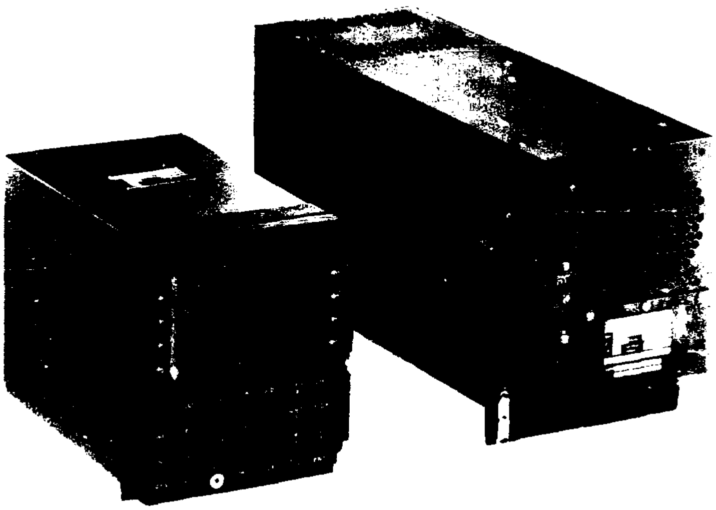
- production de logiciel par mise en parallèle de plusieurs équipes,

- parfaite adaptation aux contraintes temps réel,

- mise en place de moyens de mise au point puissants,

- documentation abondante et détaillée.

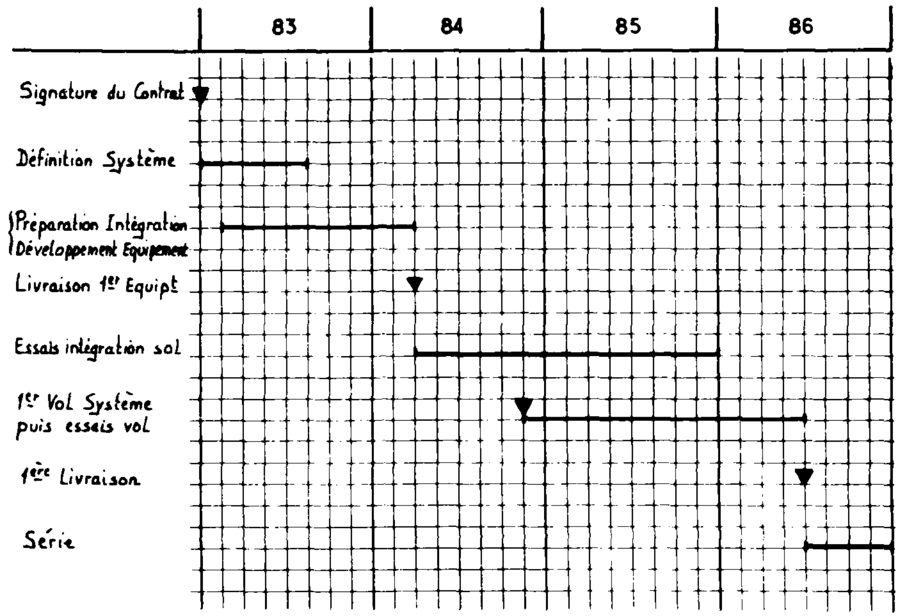
Pour la production de l'application DAUPHIN 365 F, environ 200 Koctets de logiciel ont été écrits en PASCAL ou ASSEMBLEUR.



2. DEROULEMENT DU PROJET

2.1 CALENDRIER

Le développement du système s'est étendu sur 3 ans et demi selon le planning suivant :



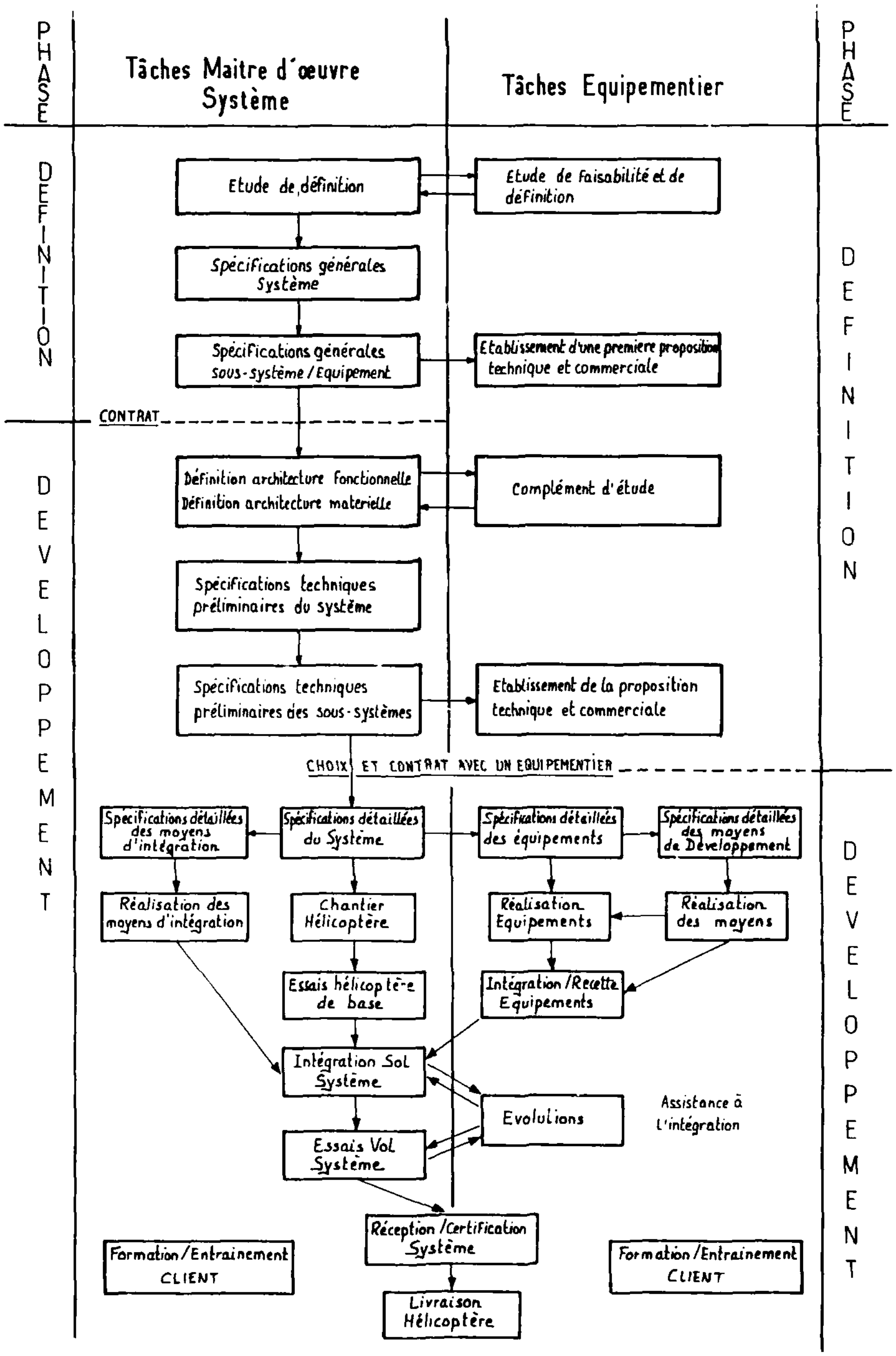
2.2 ENCHAINEMENT DES TACHES DE DEVELOPPEMENT

Le développement d'un tel système se traduit par un enchaînement de tâches successives qui mettront à contribution plusieurs services chez l'avionneur et chez l'équipementier : bureau d'études, production, assurance qualité, etc ....

L'interaction des tâches avionneur et equipementier se traduit par la nécessité :

- *d'un dialogue constant*
- *d'une grande rigueur dans l'échange des informations de définition*
- *d'une structure de travail favorisant un temps de réaction minimum.*

On peut résumer l'enchainement de ces tâches par le tableau qui suit :


PHASE
Tâches Maitre d'œuvre Système
Tâches Equipementier
PHASE
DEFINITION
Etude de définition
Etude de faisabilité et de définition
Spécifications générales Système
Spécifications générales sous-système / Equipement
Etablissement d'une première proposition technique et commerciale
CONTRAT
DEVELOPPEMENT
Définition architecture fonctionnelle
Définition architecture materielle
Complément d'étude
Spécifications techniques préliminaires du système
Spécifications techniques préliminaires des sous-systèmes
Etablissement de la proposition technique et commerciale
CHOIX ET CONTRAT AVEC UN EQUIPEMENTIER
Spécifications détaillées des moyens d'intégration
Spécifications détaillées du Système
Spécifications détaillées des équipements
Spécifications détaillées des moyens de Développement
Réalisation des moyens d'intégration
Chantier Hélicoptère
Réalisation Equipements
Réalisation des moyens
Essais hélicoptère de base
Intégration / Recette Equipements
Intégration Sol Système
Evolutions
Assistance à l'intégration
Essais Vol Système
Réception / Certification Système
Formation / Entrainement CLIENT
Formation / Entrainement CLIENT
Livraison Hélicoptère
DEVELOPPEMENT

Par ailleurs, le contrôle de la qualité intervient à toutes les étapes dès l'élaboration des spécifications, jusqu'à la réception finale de la machine.

Enfin, l'étude et le développement de moyens de maintenance peut s'effectuer, en parallèle du développement du système, en fonction des demandes du client.

2.3 MAITRISE DES EVOLUTIONS

Quelle que soit la qualité avec laquelle les spécifications ont été établies et le système réalisé, des incidents et des évolutions sont à attendre. Elles proviennent de différentes causes :

- *Des erreurs de réalisation relevées dans une configuration particulière ou dans un environnement non testé en recette*
- *Des erreurs de spécifications dues, pour certaines, à l'absence d'outils de spécifications performants permettant de décrire le fonctionnement réel du système*
- *Des évolutions de spécifications souhaitées par le client*

La prise en compte de ces évolutions conduit à la création de versions successives qu'il faut ensuite gérer correctement

Aussi est-il indipensable de maitriser ces évolutions par un processus du type de celui mis en place pour le calculateur NADIR MK2 du DAUPHIN SAR :

- *Chaque anomalie constatée chez l'avionneur ou l'équipementier après la recette, donne lieu a l'emission d'une Fiche d'Incident*
- *Ces fiches d'incidents sont analysées. Trois cas peuvent se produire :*
  - *. soit l'anomalie constatée s'explique par un contexte particulier et n'implique pas une évolution de l'équipemen*
  - *. soit l'anomalie provient d'un défaut de l'équipement par rapport aux spécifications. La modification est appliquée, et fait l'objet d'une Fiche d'Evolution.*
  - *. soit l'anomalie conduit à une évolution des spécifications. On crée une Fiche d'Evolution qui sera examinée en commun entre Avionneur et Equipementier.*

Après décision éventuelle d'application, l'évolution est réalisée, testée et intégrée à une version ultérieure de l'équipement.

Cette procédure permet une parfaite identification des diverses anomalies et une bonne maitrise des évolutions demandées. Elle permet également de bien identifier, sur le plan des coûts, l'impact de ces évolutions, et leur imputation réelle (équipementier, avionneur, ou client final)

3. MOYENS MIS EN OEUVRE

Le déroulement d'un projet de cette dimension nécessite la mise en oeuvre d'un certain nombre de moyens spécifiques adaptés aux phases :

- *de spécifications*
- *de développement*
- *d'intégration*
- *d'exploitation*

A ces moyens, se rajoute l'ensemble des moyens généraux tels que les moyens de qualification d'équipement et de simulation d'environnement.

3.1 MOYENS DE SPECIFICATIONS

La phase de spécifications revêt une importance particulière : sa bonne exécution permet de réduire les risques d'évolutions ultérieures, donc les coûts et les délais.

Elle implique la participation active de nombreux intervenants : l'utilisateur final, les bureaux d'études, les équipages d'essais en vol, etc...

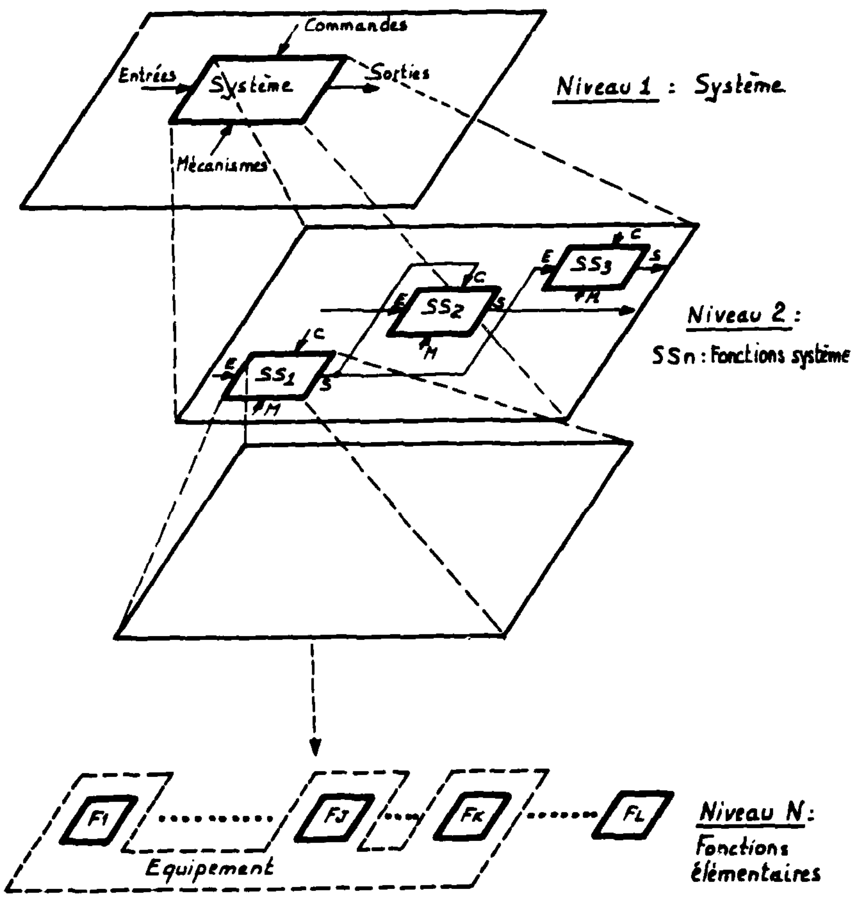
C'est avant tout une phase de communication au cours de laquelle il faut imaginer ce que sera le fonctionnement réel du système futur.

Pour ce faire, des moyens particuliers doivent être mis en oeuvre, dans le double but :

- de susciter la communication entre les intervenants
- de couvrir le plus exhaustivement possible les configurations de fonctionnement du système.

Un exemple simple, mais efficace, est l'utilisation d'un micro-ordinateur de la classe IBM PC-AT pour décrire les pages d'un poste de commande et de visualisation.

Un autre exemple, plus complexe, est celui d'un outil d'aide à la conception d'une architecture fonctionnelle de système tel que celui décrit dans le schéma ci-dessous :



La première phase du travail consiste à décomposer finement le système en décrivant les fonctions par leurs Entrées, Sorties, Mécanismes et Contrôles jusqu'au niveau souhaité.

Ensuite on procède au regroupement des fonctions élémentaires pour effectuer la projection de l'architecture fonctionnelle sur l'architecture matérielle à partir de laquelle on peut écrire la spécification détaillée des équipements (matériel et logiciel) et en particulier la spécification détaillée des traitements relatifs aux fonctions qui doivent s'exécuter dans chaque équipement.

3.2 MOYENS DE DEVELOPPEMENT

Ces moyens permettent le développement puis l'intégration et la mise au point de l'équipement.

Pour un équipement tel que le calculateur de navigation et de gestion de mission, une attention particulière doit être apportée aux moyens de développement du logiciel.

L'atelier logiciel conçu pour le calculateur ALPHA 732 est un ensemble d'outils permettant de produire et mettre au point une application dans un contexte de développement :

- multi-fonctions : une application se décompose en un lot de fonctions interconnectées

- multi-utilisateurs : une application est développée par plusieurs équipes en parallèle

Cet atelier qui met en oeuvre des méthodes de conception logicielle propres à l'aéronautique, comporte deux types d'outils :

- une chaine de production logicielle

- un système de mise au point

3.2.1 Méthodes de conception du logiciel

La réalisation d'un logiciel complexe, tel que celui d'un calculateur de gestion de mission, impose l'utilisation de méthodes de conception permettant d'une part d'assurer une production rapide, et d'autre part de faciliter au mieux les interventions ultérieures sur ce logiciel, au niveau de la phase de "maintenance".

Ces méthodes reposent sur deux principes de base :

- une description arborescente de l'application, qui à chaque fonction, associe un ensemble de tâches, et un ensemble de liens entre ces tâches, depuis l'application complète, jusqu'au niveau de modules qui constituent de véritables composants logiciels :

```
APPLICATION : Ensemble de FONCTIONS + Ensemble de LIENS,
FONCTION    : Ensemble de TACHES    + Ensemble de LIENS,
   .            .
   .            .
APPLICATION : Ensemble de MODULES   + Ensemble de LIENS.
```

Cette analyse prolonge la description arborescente qui a été faite au niveau système.

- La présence d'informations documentaires abondantes, jusqu'au niveau des modules, afin d'assurer la meilleure lisibilité possible.

Des outils logiciels spécifiques ont été créés pour assurer, en application de ces méthodes, la conception du logiciel :

. générateur d'applications (descriptif des liens interfonctions),

. outil de saisie documentaire,

. outils de simulation,

. etc...

3.2.2 Chaine de production logicielle

Il s'agit d'une chaîne de développement croisé, installée sur un calculateur hôte universel de la série VAX, sous VMS. Elle comporte une certain nombre d'outils programmés en FORTRAN ou PASCAL, qui permettent la définition du logiciel et la production de code exécutable sur ALPHA 732 :

- Editeur de texte
- Compilateur PASCAL
- Macro-assembleur
- Editeur de liens
- Générateur d'applications
- Bibliothèque de sous-programmes
- Simulateur ALPHA 732
- etc...

Chacun de ces outils dispose d'une documentation complète, ainsi que d'un jeu de tests, ce qui assure sa maintenabilité et contribue à sa portabilité sur une machine hôte (chez l'avionneur par exemple).

Cette chaîne de production logicielle est associée à un environnement de programmation assurant une meilleure gestion des développements :

- création et gestion des Fiches d'Incident et d'Evolution

- Outil de saisie documentaire

- etc....

3.2.2 <u>Système de mise au point</u>

C'est l'outil de base de la phase d'intégration matériel-logiciel. Il permet :

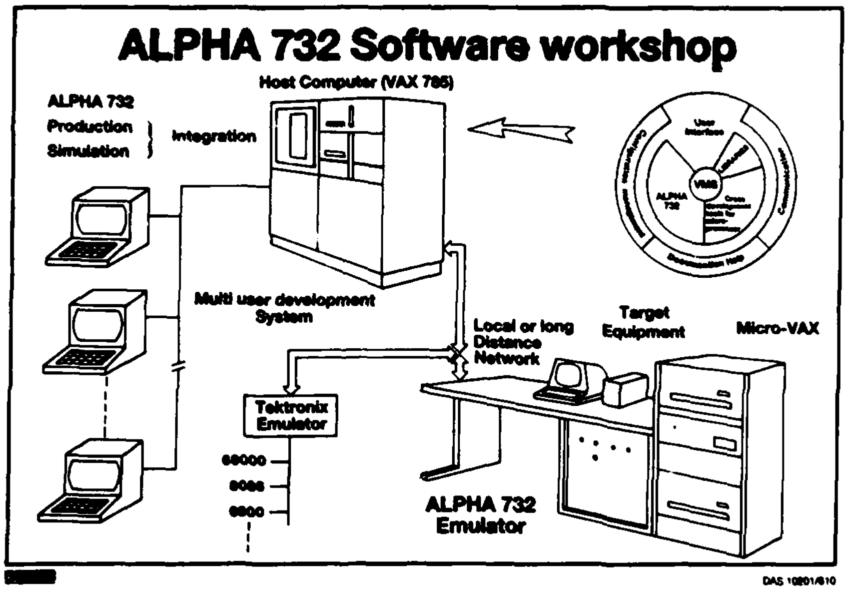
- l'émulation en temps réel
- le test et la modification des logiciels d'application
- la génération des entrées/sorties de l'équipement, simulant ainsi son environnement au sein du futur système

Cet outil, piloté par un calculateur standard micro-VAX permet une mise au point rapide des logiciels avec possibilités :

- d'observation sur points d'arrêt, en pas à pas

- de visualisation fine du fonctionnement temps réel, sans aucune perturbation du déroulement du programme

L'ensemble des moyens de développement logiciel est décrit dans le schéma suivant :



3.3 MOYENS D'INTEGRATION

Ces moyens, mis en place chez l'avionneur, permettent l'intégration des différents équipements, la mise au point du système au sol et en vol, puis la réception et une éventuelle certification de l'hélicoptère équipé.

3.3.1 Les bancs d'intégration

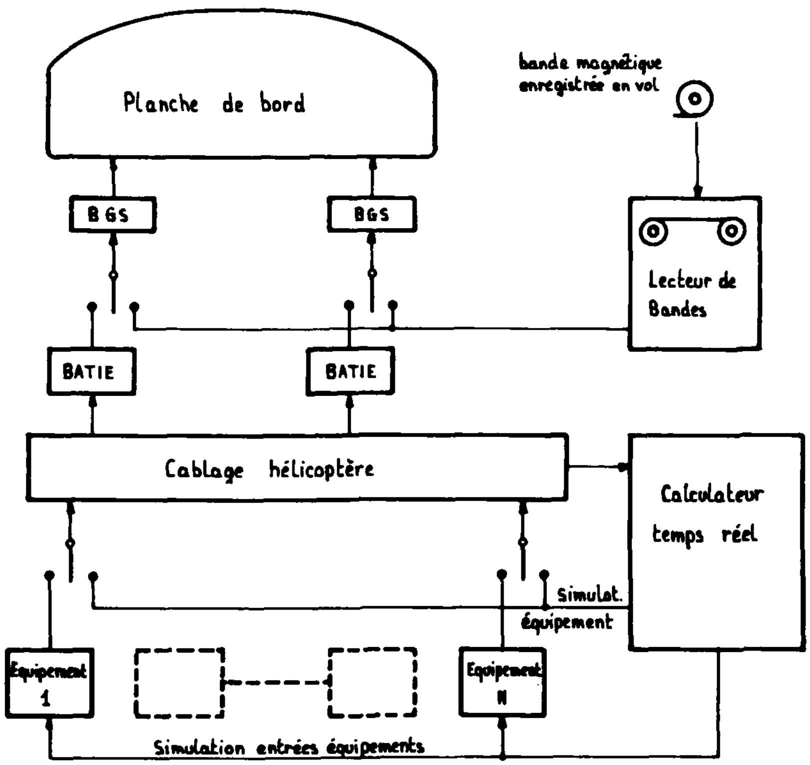
Les bancs d'intégration sont réalisés pour permettre l'intégration d'une ou plusieurs parties du système au sol en dehors de l'hélicoptère. Les principales tâches effectuées sont :

- la validation des câblages
- la validation des interfaces des équipements
- la mise au point des dialogues numériques entre équipements
- la mise au point des dialogues avec les équipages (visualisation, symbologie, commandes)
- une première validation des fonctions de chaque équipement par des stimulations statiques et dynamiques
- la reconstitution au sol, en utilisant les visualisations de la planche de bord, de certaines phases de vol ayant présenté des anomalies. Les équipes de mise au point disposent ainsi d'un moyen d'analyse au sol des phénomènes.

Le banc d'intégration mis en place pour le développement du système Irlandais était constitué :

- du câblage hélicoptère
- d'une générattion électrique
- des équipements en intégration
- de la planche de bord équipée
- d'un calculateur permettant de simuler en temps réel les équipements manquants et de stimuler le système de manière statique ou dynamique en cohérence avec des évolutions hélicoptère.

Le schéma suivant illustre l'architecture de ce banc :



3.3.2 Les hélicoptères de développement

Ces hélicoptères équipés du système sont destinés à la mise au point et la validation en vol de toutes les fonctions.

Plusieurs étapes importantes sont nécessaires :

- l'intégration du système dans l'hélicoptère qui fait l'objet d'un chantier et d'une série d'essais au sol
- le montage d'une installation d'essais comprenant des équipements de référence et des moyens d'enregistrement en vol
- des vols d'essais pour la mise au point définitive et la réception
- éventuellement des vols de certification

Ces hélicoptères de développement peuvent être, soit des prototypes, soit des appareils "société" appartenant à l'avionneur, soit les premiers appareils de série.

3.4 OUTILS DE GESTION DE CONFIGURATION

La multiplicité des fonctions réalisées par le système entraîne, à court terme, une prolifération de versions différentes, en particulier au niveau du logiciel. Il devient alors indispensable de disposer d'outils de gestion des configurations, tant chez le maître d'oeuvre système que chez l'équipementier. Ces outils font partie des moyens généraux de l'entreprise et couvrent l'ensemble des projets développés.

La gestion de configuration du maître d'oeuvre gère les états des spécifications du système ainsi que les fiches d'évolutions système. L'équipementier gère les configurations de l'équipement qu'il réalise en regard des spécifications de cet équipement elles-mêmes gérées par la gestion de configuration système de l'avionneur.

4. CONCLUSION

Ce document présente les problèmes que posent le développement d'un système avionique moderne et les solutions qui y ont été apportées dans le cadre du projet DAUPHIN SAR destiné à l'Irish Air Corps. Cet exemple met en relief les interactions nécessaires entre l'avionneur, maître d'oeuvre du système complet, et les sous-systémiers majeurs. Il souligne l'importance d'une méthodologie rigoureuse dans les échanges d'informations, de la qualité desquels dépend la bonne exécution des travaux, dans les délais et les coûts impartis. Il démontre enfin la nécessité de disposer de moyens adaptés aux différentes phases du projet, depuis les spécifications jusqu'à la réception de l'avion.

Ce système cumule à ce jour près de 1 000 heures de vol en service opérationnel et donne toute satisfaction.

Une telle expérience constitue, pour tous ceux qui l'ont acquise, un réel investissement et un gage de réussite pour le développement des systèmes futurs.

Cependant la richesse fonctionnelle et l'intégration physique allant croissant, le développement des futurs systèmes nécessitera une accentuation de la démarche et une augmentation des moyens.

DISCUSSION

W.R.Fried, US

(1) Is there any homing or direction-finding equipment on the helicopter for guidance with respect to the crash locator beacon transmissions?

(2) Why is the Decca system not used for navigation for the Irish application?

Author's Reply

(1) There is a beacon mode in the radar subsystem. The relative position of the beacon can be transferred in the navigation computer. This computer provides guidance to the location.

(2) The Irish Air Corps did not choose this system for its navigation.

G.Konomos, US

Could you please show how the NADIR MK2 is connected to and monitored by the MICROVAX to perform real-time testing with no interference?

Author's Reply

We cannot give you too many details. The main point is that, with the help of a dedicated processor included inside the bench, different observations can be made on the target equipment real-time functioning without any perturbation. The processor is never stopped. Through the connection to various probes, three traces are available: processor bus, inputs outputs, and real-time tasks election description, all events being dated precisely.

OPERATION AND PERFORMANCE OF AN INTEGRATED HELICOPTER COMMUNICATION SYSTEM

Walter R. Fried
Senior Scientist
Hughes Aircraft Company
P. O Box 3310
Fullerton, California, USA 92634

SUMMARY

The unique operational and performance requirements of the Communication System for modern tactical Army helicopters are described. An integrated system architecture is described which satisfies these requirements and incorporates very high levels of automation thereby reducing pilot workload. The automation concepts include the use of a preloaded communication data base and a centralized communication processor containing advanced control, reconfiguration and message formatting software. Link analysis and simulation results are presented which show the performance capabilities of the system with respect to the projected mission requirements.

INTRODUCTION

The Communication System of a modern tactical helicopter must satisfy several unique operational and performance requirements. Because of the current trend toward single pilot helicopters, one of the most important of these requirements is low pilot workload for operation of the communication system. Just flying the helicopter and protecting it from hostile targets and terrain obstacles will keep the pilot very busy, so that he simply will not have much time to control and operate his communication equipment. This problem becomes even more severe when the helicopter mission requires operation at Nap-of-the-Earth (NOE) altitudes, i.e., a few meters above the terrain. The latter requirement, i.e., NOE operation, is also one of the major performance challenges for the communication system design since NOE altitudes typically result in the non-existence of line-of-sight radio paths between the communicating units. Since communication is a two-way process and since different military units carry different types of radio equipment, interoperability and backward compatibility become very important considerations in a communication system. This paper describes an integrated, highly automated, helicopter communication system designed to satisfy the operational and performance requirements of future military helicopters. Typical operational mission requirements and unique performance requirements are first discussed in some detail. An integrated system architecture is described, which consists of several different equipments. The automation concepts included in the system architecture are highlighted. Results of link analyses and simulation runs for the various systems, when operating in the NOE flight environment, are presented and the performance attributes of the selected systems are described. Availability and reliability considerations of the system are addressed. Finally, the automation, integration and system design characteristics are summarized and topics for future research are indicated.

OPERATIONAL REQUIREMENTS

The requirements for an aircraft or helicopter communication system can be categorized into two types, i.e., operational and performance requirements. Although these are frequently interrelated, the operational requirements will first be discussed in this section.

Perhaps the most critical operational requirements for the communication system of a modern, military helicopter is that it be highly automated and simple to operate, in order to minimize pilot workload and human error. This is particularly important for single pilot helicopters. Currently, pilots are required to refer to Communication Electronic Operating Instructions (CEOI) books, to determine the call signs, frequencies, nets, authentication codes, OPSEC codes, etc., for the particular time period, in order to know how to call a desired destination and what frequency or channel to select. Then, the pilot needs to operate the various controls which are separately located on each radio set and which typically have different characteristics. These controls need to be highly automated, so that there are very few pilot actions required and these actions must be simple and straightforward.

If communication is suddenly lost, either through a link connectivity failure or a hardware failure, reconfiguration (i.e., change to a different radio resource) must be automatic. The pilot should merely be advised what the system has done for him.

Since the trend for modern military operations is toward more data communication, rather than only voice, the entry of data messages by the pilot must be highly automated, as compared to current practice. It is not acceptable for a busy helicopter pilot, particularly in the middle of combat, to use a keyboard to enter data, even in response to prompts appearing on displays, e.g., filling in required message fields. The entire operation must require very few actions by the pilot and be automated to a high degree. A typical example of such data communication is the automated handoff of target data from one helicopter to another helicopter, or to a ground element.

Another important operational requirement is to have a high probability that the communication link is <u>available</u> whenever it is needed. This implies high reliability of the equipment, physical redundancy of critical hardware, or functional redundancy provided by different radio systems. These automation and availability requirements lead to the need for a highly integrated architecture.

For communication systems, the problems of interoperability with fielded equipment and backward compatibility are of importance, since the operation cannot be self contained, like, say, certain navigation systems. Thus, the design of a future helicopter communication system must consider the types of equipment projected for fielding at all of the operational elements of interest, the message formats and encryption systems used, and such parameters as the expected transmitter power levels and antenna gains of ground units.

In the future battlefield, helicopters will spend a significant amount of time at NOE altitudes so that operation at these altitudes must be reliable. For self protection reasons, the transmitted signal level should be as low as possible and transmissions as short as possible, in order to minimize the probability of intercept by hostile forces. Modern military communication systems must be able to support both clear and secure voice and data traffic, and this traffic may be required simultaneously on several bands or radios. The helicopter communication system should be able to act as a relay or retransmission platform, in order to provide a means for two units who are separated by too long a range to communicate with one another.

TYPICAL MISSION SCENARIOS

In order to quantify the operational and performance requirements for a typical Army helicopter communication system, the results of previous Army studies, as well as certain projected mission scenarios, were investigated. For example, Figures 1 and 2 are derived from data in a U. S. Army Nap-of-the-Earth Communication Study. (1). Figure 1 shows the percentage of air-to-air and air-to-ground links out of 179 mission-critical paired events (i.e., transmission and acknowlegement) for an attack helicopter company operating in the SCORES Europe 1, Sequence 2A Scenario. It is seen that 92% were air-to-air links and only 8% air-to-ground links. In Figure 2, the data is further broken down, showing the range distribution, in percentage, for both the air-to-air and air-to-ground links. For example, it is interesting to note that for the air-to-air cases, 91% of all link ranges were less than 5Km, and 72% were less than 2Km.

Figure 2 shows typical communication path lengths for a composite European tactical scenario. It is seen that the postulated path lengths are quite compatible with those indicated on Figure 2, thus correlating well with the earlier U. S. Army results. Figure 4 shows typical anticipated communication links based on a projected battle team flight configuration. The links represented in Figure 4 show the need for several communication systems and nets, i.e., HF-SSB, VHF-FM, UHF-AM and PJH, in order to meet various mission requirements. For example, FM 1 (VHF-FM Net No. 1) and FM3 represent intra-platoon nets for communication between the platoon over relatively short ranges, using the VHF-FM (SINCGARS) radios. Communication between the air battle captain and the platoon leaders might be over another VHF-FM net or, if the range requires, over an HF net (HF1). Links to the ground manuever units might be over the FM2 net. Links to artillery units, such as the fire support team (FIST) and the Fire Direction Center (FDC) might be via the PJH net or an HF net. The UHF net will provide interoperability with the Air Force and the HF 2 net or PJH might be for links to the Tactical Operations Center (TOC) or Aviation Command/Control.

PERFORMANCE REQUIREMENTS

Based on the typical mission scenarios discussed in the previous sections, the major performance requirements of the communication system for future Army helicopters can be derived. Tactical air-to-air link ranges are found to be between 2 and 5KM; typical tactical air-to-ground ranges are of the order of 15Km, with some links to rear ground elements possibly being as much as 50 to 100 Km.

For high priority tactical airborne communications, it is desirable to achieve a high probability of successful communication, i.e., 0.9 or greater.

Nap-of-the-Earth (NOE) operation is a critical requirement and the NOE altitude has been defined as 3M above the terrain. Types of terrain encountered in tactical operations range from rough, wooded, mountainous terrain to more gentle, hilly terrain. Ambient radio frequency noise varies from the moderate rural areas to the more severe urban areas.

SYSTEM ARCHITECTURE

In order to meet the operational and performance requirements described in the previous sections, a highly integrated communication system architecture was evolved, as shown in Figure 5. It includes a central communication processor which is the "brain" of the system. The processor software provides the initialization, control and reconfiguration of all of the radio functions and of the audio control unit and it also performs the message processing required for data communications. The processor is interconnected to the various radio functions via local multiplex data busses. Typically, two busses are required, one for control data and one for message data. The Communications Processors acts as the bus controller for these local busses. As shown in the figure, a centralized Control/Display in

the crew station is used by the pilot to perform any communication control functions. The interface between the Communication Processor and the Control/Display, as well as with the other avionics system, is via the global avionics bus.

To meet the performance requirements outlined earlier, a variety of communication equipments are included. Because of the importance of operation at NOE altitudes, an HF-SSB radio function operating in the 2-30 MHz frequency band with ECCM capability, is included. The HF-SSB radio will support ground wave propagation to a significant range; near vertical incidence skywave (NVIS) is applicable for medium range requirements and normal skywave for the very long ranges. The frequency hopping (ECCM) capability is required to provide anti-jam performance. The HF function supports voice and data traffic in both clear and encrypted modes of operation.

The VHF-FM (SINCGARS) radio function operates in the 30-88 MHz band and provides interoperability with a large number of U. S. Army elements. It supports NOE operation to a more limited range than the HF radio. It includes a frequency hopping (ECCM) mode to provide anti-jam performance. It has both voice and data capability, at a data rate up to 16 Kbps. In the data mode, it is compatible with the Army's TACFIRE network. The VHF-FM (SINCGARS) function is included in a dual redundant manner, in order to facilitate simultaneous dual net operation and to provide a retransmission capability. In order to enhance range performance, particularly at NOE altitudes, a high power VHF-FM amplifier is included.

The VHF-AM radio function operates in the 116-152 MHz band for voice communication and is included to provide air traffic management and backup air-to-air capabilities.

The UHF-AM (HAVE QUICK) radio function operates in the 225-400 MHz band and primarily provides interoperability with the Air Force for voice communication and military air traffic management operations. It includes a frequency hopping anti-jam mode.

The PJH Enhanced PLRS User Unit (EPUU) provides a highly jam-resistant, secure, direct user-to-user data communication capability, in conjunction with the PLRS portion of the PLRS/JTIDS Hybrid (PJH) network. It operates in the 420-450 MHz band and uses two spread spectrum techniques, i.e., frequency hopping and direct sequence spreading, with short burst transmissions. The PJH system also provides inherent automatic relay and net management capabilities. Optimum relay paths are automatically determined by the system, based on link quality measures. The EPUU formats the data used in normal TACFIRE messages in an efficient data format. The current data rate capability is 1200 BPS and is in the process of being increased further through the insertion of VHSIC technology. As an example, PJH communications is particularly useful for target data transfer from the FAAO to artillery elements, such as the FIST and FDC, since the longer ranges required for these links can be easily achieved through the PJH relay capability. For operation beyond the forward line of troops (FLOT), the high stability of the EPUU clocks permits communication to be maintained for a long period, even after connectivity with the PJH Net Control Station has been lost. Accurate position information of all elements is continuously available within the PJH network so that the EPUU can provide own-position updates to the helicopter's navigation system.

The Audio Control Unit (ACU) routes the voice audio from/to the pilot's microphone/headset and the radios. As shown in the architecture diagram in Figure 5, the control of the ACU is provided by the Communication Processor via the local multiplex control bus.

The IFF Transponder is included to provide self protection of the helicopter against attack by friendly forces. The IFF Interrogator is included so that potential airborne targets can be interrogated, in order to avoid attacking friendly aircraft. Both, the IFF Transponder and Interrogator are also controlled by the Communications Processor via the local multiplex data bus.

AUTOMATION CONCEPTS

One of the primary goals of the design of the communication system was to minimize the pilot workload associated with its operation. It became clear that the pilot should have no involvement or very little involvement in the initialization, configuration and (in case of failures) reconfiguration of the communication system. Similarly, the required pilot actions for entry of message data, message formatting and message reception should be limited to as few as possible. Toward this end, a maximum level of automation was included in the design and these automation concepts are described in this section.

In operation, before the start of a mission, a communication pre-mission data base is loaded into the processing system memory. This data base includes the electronic CEOI data applicable for the geographic area of interest, such as the nets, frequencies, call signs, authentication codes, OPSEC codes for the various Army organizational elements, for different time period of the day. In addition, the data base will include a mission-unique Communication Plan (COMM PLAN), which includes the units with whom communication is anticipated, the primary and alternate radio bands and the types of nets, such as voice or data, which are to be used during the mission. After appropriate conversion to the radio parameters, the Communications Processor software then automatically initializes the various radios at the time of takeoff.

At any time thereafter, the pilot uses the centralized control-display to initiate a communication event. For example, using his display, the pilot may select from a matrix of symbols, representing force elements, the particular unit with which he needs to communicate. Alternatively, a voice command system may be used for this action. The Communication Processor (CP) software through use of the pre-stored data base, then causes the applicable call sign, primary and secondary band authentication code, etc., to be displayed to the pilot. At the same time, the CP automatically configures the primary radio for the channel or net applicable at that time. As a result, the system is immediately ready for the pilot to initiate a voice transmission. The capability for a manual control override by the pilot must also be available.

If the transmission of a digital data message is desired, the pilot would designate the desired destination in the same manner as described previously for voice traffic and then select the data mode of operation. Again, the proper radio resource has been selected and configured by the software. The data messages can then be entered and transmitted in an automated manner. For example, for a target handoff function, the pilot merely initiates and commands the process by selecting the recipient and the specific target. The software automatically performs any required coordinate conversion for the target position, formats the required data into the appropriate message format, (e.g., the TACFIRE format), and routes the message to the selected radio over the data bus. Acknowledge processing is used to sense any link connectivity failures. In case of such a link failure (for example due to lack of line-of-sight), the software automatically reconfigures the system to an alternate radio resource. Similarly, if the built-in-test (BIT) functions sense a hardware failure in a particular radio, the software automatically reconfigures the system, based on the information in the pre-loaded COMM PLAN and CEOI data base. The pilot is informed (or alerted) of the reconfiguration actions taken via his control-display, but he does not need to get involved in any action. Thus, the automation concepts included in the system design provide for control and operation of the Communication System with a minimium involvement by the pilot, therefore adding little to the total pilot workload.

PERFORMANCE ANALYSIS RESULTS

Link analysis and simulation runs were conducted in order to determine the performance of the selected communication system with respect to the mission scenarios described earlier. Specifically, the achievable range performance of the three primary tactical radio systems was analyzed as a function of terrain, ambient noise and probability of successful communication, and the results were compared to the requirements discussed earlier.

The link analysis and simulation effort made use of the Transmission Simulation Program (TSP) which had previously been developed by Hughes Aircraft Company. It facilitated the parametric analysis of message error rate versus link distances for the various systems being investigated, as a function of modulation waveform, message structure, data rate, terrain, ,etc. These data were then used as the basis for further link analyses, giving achievable ranges and signal margins, as a function of transmitter power, antenna gains and different link geometries. For most of the analyses, an NOE altitude of 3m and a probability of successful communication of 0.9 were assumed. Both, rough mountainous terrain (which might be typical of the Fulda gap in Germany) and more moderate, hilly terrain, such as might be typical for certain mid-east areas, were analyzed. Urban ambient radio noise levels and more moderate rural noise levels were treated. Both air-to-air and air-to-ground communication links were analyzed.

Figure 6 shows a comparison of achievable air-to-air communication ranges for the three primary tactical Army radio systems analyzed versus a histogram of the required radio transmissions, as derived from the data in Reference 1. It is seen that all three of the systems will meet the bulk of the required link ranges in the worst type of terrain, although HF radio out-performs the other two systems. Figure 7 shows a similar performance comparison for the projected air-to-ground links. For this case, only HF radio and multi-hop (relay) PJH links satisfy typical helicopter-to-artillery element link ranges.

Similar link analyses were performed for a jamming environment, but are beyond the scope of this paper. The results of that analysis revealed that frequency hopping (ECCM) techniques are absolutely essential in HF and VHF-FM radio systems in order to provide the required communication performance.

In summary, the results of the performance analyses indicate that the selected communication system meets the tactical range requirements for the mission scenarios described earlier. Use of the higher available transmitter powers and use of ECCM in HF and VHF-FM radios are required. For medium range requirements, the use of relay (for example with PJH) or a retransmission mode will be needed. HF radio NVIS and conventional skywave propagation modes will be used to meet the longer range requirements.

To achieve the greatest possible mission critical reliability performance, both physical and functional redundancy are employed. Examples of physical redundancy are the use of dual VHF-FM (SINCGARS) functions, dual power supplies, dual transmitters, etc. However, use of "functional" redundancy is often equally or more important than "physical" redundancy. The reason for this arises from the fact that if the environment or transmission medium have caused the link to fail, redundant hardware will not solve the problem. Typical examples of this are loss of line-of-sight due to terrain features for VHF and UHF systems and ionospheric disturbances or atmospheric noise for HF systems.

CONCLUSIONS

An integrated helicopter communication system has been described, which is highly automated, thereby relieving the pilot of the workload required in current systems for the control and operation of distributed radios and for the entry of data messages.

Link analysis results have shown that the system meets the operational and performance requirements which have been projected for tactical helicopter mission scenarios of the future, notably for operation at NOE altitudes.

There are several areas which deserve attention for future research and development. For example, in order to ease field operations, a simple method for centrally loading security codes into COMSEC equipment needs to be developed. All of the COMSEC/TRANSEC functions should be fully embedded within the radio hardware and effort is required to develop the interface and control functions to accomplish this.

Millimeter wave radio equipment operating in the 54-60 GHz oxygen absorption band can provide a unique covert communication capability, i.e., an extremely low probability of intercept. Therefore, inclusion of that type of system in the future, either as a separate function or possibly as an applique to another radio, should be investigated.

ACKNOWLEDGMENTS

The author wishes to acknowledge the contributions made by E. Larsen of Hughes Aircraft Company in the areas of operational requirements and link performance analysis.

REFERENCES

1. "Nap-of-the-Earth Communications, Concept Formulation Package", U. S. Army Aviation Center, Ft. Rucker, Alabama, 1979, Vols 3 and 5.

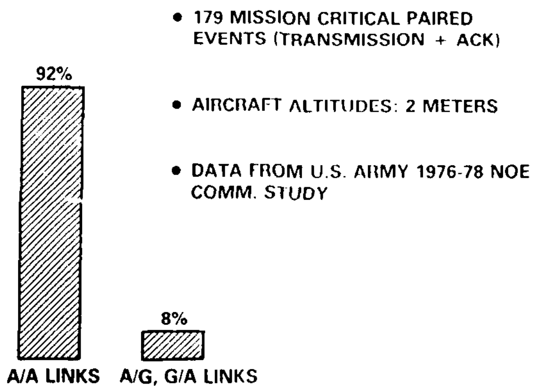


FIGURE 1. COMMUNICATION EVENTS FOR ATTACK HELICOPTER COMPANY, SCORES EUROPE 1 SEQUENCE 2A SCENARIO

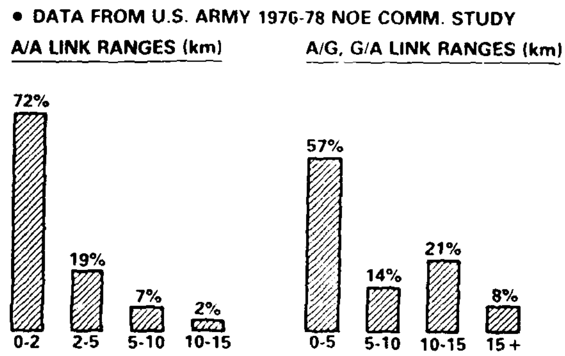


FIGURE 2. RANGE DISTRIBUTION OF COMM. EVENTS FOR ATTACK HELICOPTER COMPANY SCORES EUROPE 1. SEQUENCE 2A SCENARIO

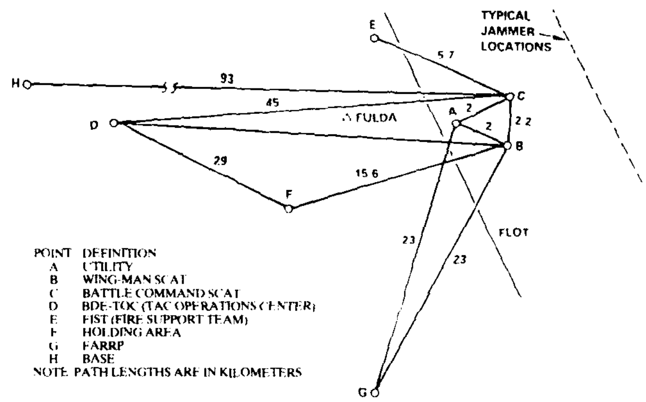


FIGURE 3. COMPOSITE EUROPEAN SCENARIO COMMUNICATION PATH LENGTHS

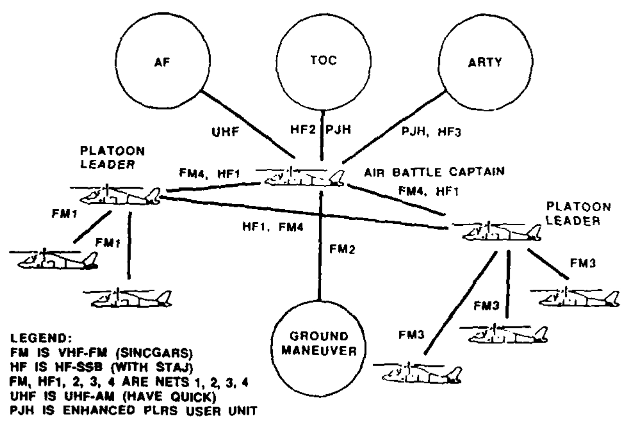


FIGURE 4. TYPICAL TACTICAL COMMUNICATION LINKS

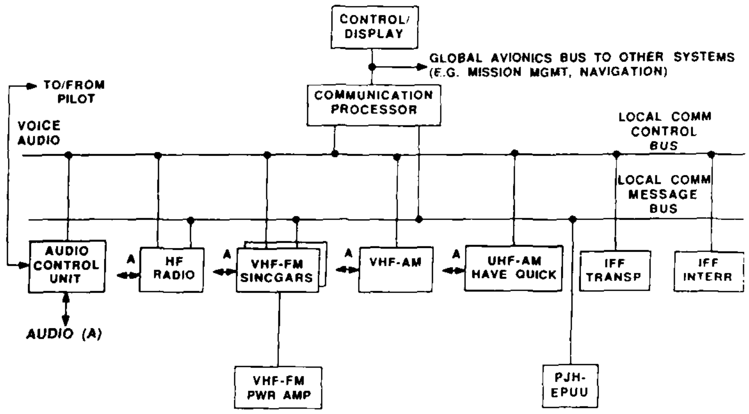


FIG. 5: COMMUNICATION SYSTEM ARCHITECTURE

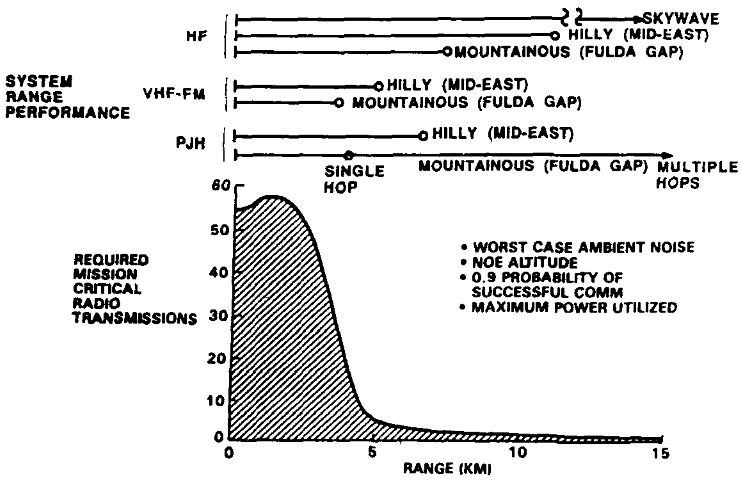


FIGURE 6. AIR TO AIR COMMUNICATION RANGE PERFORMANCE VERSUS REQUIREMENTS

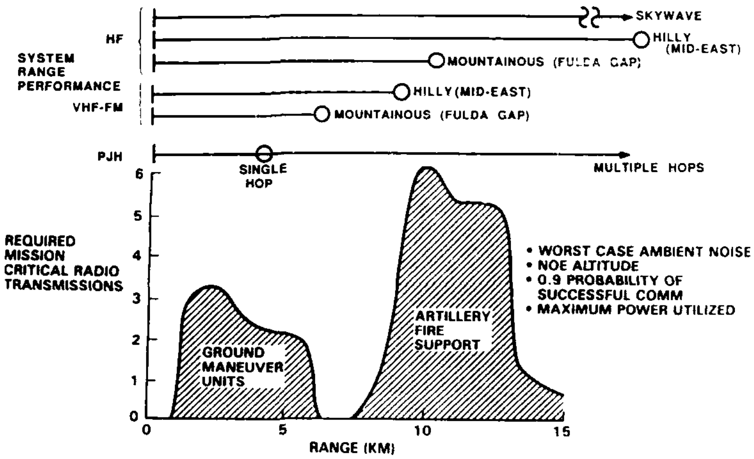


FIGURE 7. AIR-TO-GROUND COMMUNICATION RANGE PERFORMANCE VERSUS REQUIREMENTS

DISCUSSION

D.W.Hussey, UK

To what extent are digital terrain elevation databases relevant to NOE battlefield communications, particularly with regard to LOS prediction?

Author's Reply

The digital terrain elevation database could be used in a communication processing algorithm to determine the optimum radio band resource and/or the optimum relay paths to complete a desired communication link to an element.

G.M.Barling, UK

In your paper, you claim:

- Probability of successful communication is greater than 0.9.
- The system automatically reconfigured following failure with no manual intervention.

Does the probability figure include an allowance for undetected failures?

Author's Reply

The "0.9 probability of successful communication" value was used in the link analysis, from which results were presented. Analysis has not yet been made on the effect of undetected failures on the probability of successful communication.

M.Kayton, US

I suggest that Figures 6 and 7 show the number of transmissions per range band. As it is, an infinite number of transmissions are called for.

Author's Reply

The current figures are intended to show at which ranges required radio transmissions are most likely. They were derived from actual histograms of transmissions in certain radio bands o. the type shown in Figure 2.

J.A.Salmon, FR

The IFF is included in your communication system (Figure 5). How does it participate in this system?

Author's Reply

The communication processing function initializes the IFF transponder and interrogator using preloaded security codes located in the processing system memory. Power control of both functions is executed through the communication processor. Interrogator triggering can also be accomplished through the communication processing function.

DESIGNING FOR DESIGN EFFECTIVENESS OF COMPLEX AVIONICS SYSTEMS

by
Kenneth R. Boff, Ph.D.
Human Engineering Division
Harry G. Armstrong Aerospace Medical Research Laboratory
Wright-Patterson Air Force Base, Ohio 45433-6573, USA

SUMMARY

Reliable data on the aircrew's ability to acquire and process task-critical information are of prime importance to the design of effective controls and displays. While the available body of psychophysical research contains a staggering volume of human perceptual and performance data and principles that are of potential value to the design process, these are not systematically considered in the typical design of avionics systems. Though the nature and availability of these data are a key part of this problem, it can also be attributed to the basic skills and inclinations of designers, limitations in the available support environment, and constraints imposed by the system design and acquisition processes.

Complex system design may be characterized as a creative integration and/or skillful blending of technologies counterbalanced to accomplish a predefined function within material, cost and schedule constraints. Design effectiveness is a function of the cumulative "goodness" of design decisions and tradeoffs which collectively meet design requirements within these constraints. System effectiveness depends on design effectiveness but may var with changing demands and user perspectives in the course of system deployment. A common denominator in system design is "information" and the efficiency with which it is factored into design decisions and tradeoffs. The use of information, however, varies as a function of its "perceived" value and cost (in terms of risk/time dollars) which may be independent of its "real" value and cost.

The Integrated Perceptual Information for Designers (IPID) Project is a multiagency supported effort to aid the accessibility and use of human performance data in system design. It is formulated around five information management objectives geared toward: 1) Identifying, collecting, and consolidating human performance data of potential value to system design; 2) Human factoring these data to enable their direct use by system designers; 3) Establishing an institute with responsibility for maintenance, update and analysis of these resources to support crew system design; 4) Developing and sponsoring of educational opportunities to train and sensitize system designers in the value and application of human performance data to crew system design; 5) Conducting exploratory research to define and evaluate requirements for an automated design support capability to aid designers to efficiently access and trade off human performance data with other technical information germane to the effective design of crew systems.

I. INTRODUCTION

Conducting modern warfare, particularly against numerically superior forces, demands that Air Force research and development programs support the design of avionics systems that are maximally effective. However, despite an infusion of new display and data-handling technologies into fighter cockpits, aircrew workload continues to be a problem. State-of-the-art crew systems present a staggering volume of codified visual and aural information which competes for the pilot's attentional resources (see Figs 1 and 2). While few would disagree with the contention that the operator's ability to acquire and process task critical information is a major contributor to system effectiveness, the design of crew systems has typically involved little systematic consideration of human performance characteristics and limitations. While a good deal of potentially useful human performance data exists, these data have had little direct impact on the design of crew system interfaces. Though the nature and availability of these data are a key part of this problem, it can also be attributed to the basic skills and inclinations of designers, limitations in the available support environment, and constraints imposed by the system design and acquisition processes (1, 2).

II. RELATIONS BETWEEN THE USE OF TECHNICAL INFORMATION IN SYSTEM DESIGN AND DESIGN EFFECTIVENESS

Though human performance data or other information germane to a given design must exist before it can be used, availability of information is not a sufficient condition to ensure its use. For information to be used, it must have some perceived value or relevance to the problem at hand. Domains of technical information believed not to be of substantive value will not be accessed regardless of the real value of the information. Perceived value, in turn, is a function of the prior training and experiential biases of the individual user.

Independent of perceived value, the functional value of pertinent technical data to a given design decision may be less than that implied when considered in a broader system design context. The design of an aircraft system typically involves balancing tradeoffs among many subsystem priorities to achieve a required "functionality" within material, cost, and schedule constraints. Performance optimization of the human-avionics interface is but one of many competing design goals. In aircraft system design, other subsystem goals may take precedence in tradeoffs with crew system capabilities. For example, aerodynamic requirements will influence windscreen shape which will determine crew system volume which, in turn, can compromise crew subsystem design decisions. Therefore, satisfying, rather than optimizing, subsystem goals may be optimal strategy in the context of a given system design.

In practical terms, "effective" systems are those which operate efficiently (i.e., perform the function for which they were designed), reliably (i.e., minimal unpredictable failures and ease of supportability), and competitively (i.e., costs to acquire, operate and maintain over the system life cycle). For crew systems, in which the human is a vital component, operability (i.e., functional match between user and system capabilities) and acceptability (i.e., users expectations of and willingness to use the system) are major factors contributing to a system's "effectiveness." Effectively designed systems, therefore, require some consideration of the factors which support these attributes.

Whether a given system is perceived as "effective" is also driven by factors outside of the control of the designer which, nonetheless, must be planned for. To the extent that operational mission needs change and new applicable technologies emerge, systems become obsolete as a result of their lowered operational effectiveness. User acceptance, which may itself be manipulated by marketing and sales tactics, will also influence perceived effectiveness. In military system acquisition, it is the customers and end users who define requirements and control the training, operations, and maintenance environments which influence the perceived effectiveness of a given system long after it has been designed.

The interdependence among the myriad variables which contribute to the design of complex systems, including such behavioral intangibles as creativity (3), makes it difficult to predict the influence of any single factor on a given design. In contrast, the pressures of limited time and resources typical in system design drive designers to bias decisions and tradeoffs towards reduction of uncertainty or risk. Hence, it is not surprising that "new" crew systems typically are "adaptive" designs (i.e.,adapting a known solution to a changed task) or "varient" designs (i.e., varying parameters such as size, arrangement, or timing of a known design solution without changing the basic design) as opposed to "original" designs which may depend on untested approaches or technology (see reference 4 for a discussion of "adaptive" and "varient" designs). The selection of an appropriate baseline -- a "proven" system or subsystem design analogous to the one under development -- will generally account for the largest portion of variance in a given design's effectiveness. In other words, selecting a good baseline match should reduce risk to system effectiveness. Effective baselining of a system design may itself be constrained by a lack of relevant past designs (e.g., prototyping the first lunar lander), inaccessibility of proprietary data, or corporate biases. As a system design strategy, baselining is employed in one form or another throughout system and subsystem levels within any given design. Theoretically, "original" design decisions should then precipitate from consideration of those aspects of the baseline which are either not responsive to current system requirements or are not effective in meeting these requirements. However, it is often the case in military system acquisition that system requirements and design decisions are biased toward potential opportunities posed by the availability or near-availability of new technologies which, in turn, adds risk to achieving design effectiveness. Effective design, therefore, involves a skillful blending of past baselines with new decisions and tradeoffs, counterbalanced to minimize risks to achieving predefined functionality within material, cost and schedule constraints.

Raising the efficiency with which information is considered and factored into design decision-making should, by inference, raise the probability of "design effectiveness." Conversely, design decisions made without consideration of potentially leveraging information may be sub-optimal and may collectively, depending on their impact on system function, undermine design effectiveness (1). Ironically, enhancing design effectiveness by improving designer access to and utilization of design relevant information may be hampered by the fact that designers are already deluged by too much information competing for their time and attention. Given these conditions, designers will seek new information only to the extent that it has high perceived value (i.e., may reduce risk) and low perceived cost in terms of the time and resources to acquire and interpret (5). In other words, it is unlikely that additional information will be sought beyond that perceived as satisfactory.

Hence, the design of controls and displays effectively matched to the capabilities of the pilot is dependent on a complex set of factors bearing on the access to, handling of, and decision making with information. To the extent that sound data exist and are used, these may enhance the probability that a given system will be effective, though this can never be fully assured. In this context, designing for design effectiveness implies aiding the accessibility and use of information in design decision making.

III. A CONCERTED APPROACH TO DESIGN INFORMATION MANAGEMENT: THE INTEGRATED PERCEPTUAL INFORMATION FOR DESIGNERS PROJECT

Recognizing the need to improve effectiveness of the human-avionics interface in crew systems, the Harry G. Armstrong Aerospace Medical Research Laboratory set up a project, in 1980, to develop a sound theoretical and empirical basis for human-centered crew system design. The principal assumption of this project, known as TACDEP (Tactical Aircraft Cockpit Development and Evaluation Program), is that crew system effectiveness ultimately depends on optimizing the match between the bandwidth of controls and displays and the sensory, perceptual and cognitive characteristics of the human operator. While this assumption is difficult to prove, it is a reasonable inference based on lessons learned from operability and workload problems reported in fielded systems and accident investigations. In addition, though it is generally agreed that the operator's ability to acquire and process task critical information is a key contributor to system effectiveness, few resources exist to support designing for these operator functions. Existing data of potential value to design are widely scattered and difficult to discriminate within the voluminous research literature. Furthermore, even with germane data in hand, it may be difficult for non specialists to assess the data's relevance to their problem (1). The problem is further exacerbated by the low perceived value of these data by the design community which is fueled by both negative experiences with these data (6, 7) and the pressure to account for a myriad of other system-related variables in the course of crew system design.

Based on this characterization of the problem, the Integrated Perceptual Information for Designers (IPID) project was set up to support TACDEP. IPID is a multi-agency effort supported by organizations within the U.S. Air Force, Army, Navy, and NASA. Its prime objective is to provide "high-value" human performance data as a "low-cost" resource to designers of operational crew systems and training devices. The project is organized around five interrelated information management and product objectives:

1) Consolidation of potentially useful human performance data;

2) Presentation of these data in a form useful to system designers;

3) Real-time support for Analysis of the existing data with respect to system design problems;

4) Training and sensitization of system designers of the value and use of human performance data in system design;

5) Support Accessibility and application of human performance data in context with other design relevant data.

Collectively, these objectives, discussed in detail below, are aimed at raising the perceived value and lowering the perceived cost of using human performance data in system design.

1. CONSOLIDATION

The first objective of the project was to identify, collect, and consolidate the existing human performance data of potential value to the design of operator control and display interfaces. This involved the design and development of a professional level reference work involving detailed treatment of forty-five subject areas (see Table 1) by over sixty recognized experts. The resultant Handbook of Perception and Human Performance (Boff, Kaufman, and Thomas, Eds; 8, 9) is a two-volume work of approximately 2800 pages. The Handbook differs from standard texts in its emphasis on self-contained units of information, detailed indexing and cross-referencing. It is illustrated with over 1600 figures and makes extraordinary use of data functions and schematics to present technical material. Data are plotted in standard units based on the Systeme Internationale (10). Figures, tables, and their captions are designed to "stand alone" so as to be interpretable independently of the text. For example, the captions provide a description of variables evaluated in the reported study, an indication of the data reliability, a "bottom line" summary of what the data are about, and a reference giving the source of the data. While the intended user of the Handbook is the ergonomist or R&D engineer, it provides a reliable basis for subsequent products of more direct design relevance.

Table 1 - Handbook Table of Contents

Section I: THEORY AND METHODS

Table 1 - Handbook Table of Contents - Continued

---

2. PRESENTATION

This second objective is concerned with development of an approach to communicating ergonomics data to system designers who, while having little prior training and experience with ergonomics, need reliable data to support design decisions or tradeoffs between human performance and equipment/environmental considerations. The Engineering Data Compendium (11) is a reference document which consolidates human sensory/perceptual and performance data in a form human-factored for system designers. It provides comprehensive and detailed specifications on the capabilities and limitations on the human operator, with special emphasis on those variables which affect the operator's ability to acquire, process, and make effective use of task critical information.

Information was selected for inclusion into the Compendium on the basis of its practical potential for system design through an iterative process of review and

analysis employing hundreds of technical subject matter experts and "designers." Prospective entries were reviewed on the basis of statistical and methodological reliability, applicability to the normal adult population, and potential relevance to design problems. The Compendium consists of approximately 1200 concise two page entries (see Figure 3) designed to be self-contained, with information from related studies summarized and presented in graphic form wherever possible. Entries have been prepared treating parametric data, models and quantitative laws, principles and nonquantitative laws. Expected to be published by the Air Force (11), it will consist of four volumes -- three looseleaf volumes of perception and performance data and a bound User's Guide containing supplementary aids such as instructions for locating and using individual entries, design checklists, indexes, and a glossary. Specific programmatic details of this effort are summarized in Reference 12.

3. ANALYSIS

In conjunction with the Tri-Services and NASA, the Harry G. Armstrong Aerospace Medical Research Laboratory will establish and host, beginning in 1988, a Crew System Ergonomics Information Analysis Center (CSERIAC). CSERIAC will provide a full range of technical information services in support of crew systems research, design, and development in the government, industrial and academic sectors. The essential mission of CSERIAC is to maintain contact with the relevant knowledge and experience bases across these sectors and to develop the ability and media to draw upon and focus this expertise to solve problems, achieve expert consensus and aid planning for more effective use of ergonomics data in the system design process. In addition, CSERIAC will provide information services, including topical reviews, special analysis reports, data, models, design support software, methodological assistance and a "Current Awareness" newsletter. Maintenance and update of data bases, including the IPID Engineering Data Compendium will also be a function of CSERIAC.

To determine the validity of the need for a DOD center devoted to the analysis and dissemination of crew system ergonomics information, 3705 potential users within DOD and industry were surveyed by a mail questionnaire (13). Eighty-seven percent (87%) of the 829 respondents agreed that a Department of Defense CSERIAC was the appropriate mechanism to meet this need. Seventy-nine percent (79%) of the respondents work in research and development, management or design. Ninety-seven percent (97%) use crew systems ergonomics information. Seventy-eight percent (78%) are willing to pay fees for CSERIAC services. Over 4000 requests per year for CSERIAC services would be made by the survey respondents alone.

The major initiating task for CSERIAC will be to achieve the credibility and aura of a "center of excellence" capable of attracting the range of professional support across the international community essential to its usefulness and long-term survivability.

4. TRAINING

This objective of the project is to enhance the perceived value and demonstrate the applicability of ergonomics data for system designers through a series of educational/training opportunities. The first of a proposed series of short courses and seminars was conducted under the auspices of the University of Dayton in Dayton, Ohio during 8-13 June 1986. Offered as the "Human Perception and Performance Workshop for System Designers," its primary goal was to provide system designers with a human performance framework for "decomposing" equipment-related design problems while sensitizing participants to the issues and approaches in using human performance data in human-machine system design. Future training functions are in planning and include development of an AGARD Lecture Series.

5. ACCESSIBILITY

The fifth information management objective of the project is to aid the accessibility and application of human performance data in the context of other system design-related information and procedures. "Designer's Associate" is a four-year exploratory research program to define and validate functional specifications for a "human engineered" design support system which efficiently services the integrated technical information needs of operational and training crew systems designers.

The program approach, illustrated in Figure 4, is fundamentally based on testing IPID project assumptions regarding problems and issues in the handling of technical information by system designers. This has engendered the need to develop an integrated understanding of the role and handling of technical information in the process of designing operational and training crew systems across Government and industrial sectors. Field interviews with system design personnel help to identify common needs, problems, and perceived benefits in their use of technical information and its implications for design products (14).

Information handling issues, deemed within the scope of Designer's Associate (DA), are analyzed in terms of causes and implications. Based on this analysis, candidate DA support functions and capabilities which have potential to resolve these problems are identified; these functions are, in turn, analyzed to determine the supporting technologies necessary for their implementation. Those functions which can be supported by current technology or prospectively supported by emerging technologies

(e.g., machine intelligence; 15), collectively comprise the Designer's Associate system concept. This concept will be validated and documented at individual function and integrated system levels through a series of tests and evaluations involving simulation, laboratory studies, field studies, interviews with designers and design management, and development of research software. Research to advance "borderline" technologies will be supported in diverse areas including cross-disciplinary access and interpretation of technical information, decision aiding, and user-system interface. The final products of the program will be a series of demonstrations of Designer's Associate functions and capabilities and a functional system specification sufficiently validated to support justification for advanced development.

Production of the Designer's Associate support system by the mid-1990's should facilitate information handling and decision making for crew system designers. Designers will have greater potential then ever before to draw upon available knowledge resources, presently widely diffuse, and focus these consistently into all levels of design decision making. This capability coupled with decision aiding and advanced user interface technologies will, in turn, provide the power to rapidly consider more alternatives, more effectively, thereby enhancing the probability of system effectiveness.

IV. CONCLUSIONS

Design effectiveness is the degree to which system function meets design requirements within cost and schedule constraints. It is a function of the cumulative goodness of design decisions and tradeoffs, which are, in turn, dependent on the information factored into these decisions. Serendipity excluded (e.g., the resistor design that makes a fine lightbulb), design effectiveness is a necessary though not sufficient condition for system effectiveness which may vary with factors outside of the control of the designer and design process (i.e., changing demands or requirements of the operational, maintenance and training environments in which the system is deployed). Hence, while the probability of system effectiveness may be enhanced by design effectiveness, it can never be ensured.

The products of the IPID Project will support the avionics system designer's ability to match crew system displays and controls to the performance capabilities of the operator. The Handbook of Perception and Human Performance (8, 9) and human Engineering Data Compendium (11) consolidate and package these data in a comprehensible format. CSERIAC will provide access to the state-of-the-art in crew system ergonomics through interactions with current experts and analysis of the literature. It will influence both the definition of requirements and the details of design necessary to accomplish them. Professional short courses, conferences, and symposia will sensitize and familiarize system designers with the value of these data while lowering perceived risks in their use. The Designer's Associate support system will eventually automate access and utilization of these data in the context of other relevant design information and aid the designer to factor these into design decisions and tradeoffs, thereby contributing in a meaningful way to design effectiveness.

REFERENCES

1. Boff, K.R. The Tower of Babel Revisited: On Cross-disciplinary Chokepoints in System Design. In System Design: Behavioral Perspectives on Designers, Tools and Organizations. 1987. Elsevier, New York.

2. Rouse, W.B. and Boff, K.R. System Design: Behavioral Perspectives on Designers, Tools and Organizations. 1987. Elsevier, New York.

3. Rouse, W.B. A Note on the Nature of Creativity in Engineering: Implications for Supporting System Design. Information Processing and Management. 1986. 22. 279-285.

4. Pahl, G. and Beitz, W. Engineering Design. 1984. Springer-Verlag, New York.

5. Rouse, W.B. On the Value of Information in System Design: A Framework for Understanding and Aiding Designers. Information Processing and Management. 1986. 22. 217-228.

6. Meister, D. and Farr, D. The Utilization of Human Factors Information by Designers. 1966. Office of Naval Research, NONR Contract #4974-00.

7. Klein, G.A. and Brezovic, C. What do Training Device Designers want to know about Human Performance Characteristics? Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio. AAMRL-TR-87-XX, In Press.

8. Boff, K.R., Kaufman, L. and Thomas J., Eds. Handbook of Perception and Human Performance. V.I: Sensory Processes and Perception. 1986. John Wiley and Sons, New York.

9. Boff, K.R., Kaufman, L., and Thomas, J., Eds. Handbook of Perception and Human Performance. V.II: Cognitive Processes and Performance. 1986. John Wiley and Sons, New York.

10. U.S. Department of Commerce: National Bureau of Standards. The International System of Units (SI). 1977. NBS Special Publication 330. Washington D.C.

11. Boff, K.R. and Lincoln, J. Engineering Data Compendium: Human Perception and Performance. (4 Volumes). Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, OH. In Press.

12. Boff, K.R., Calhoun, G.L., and Lincoln, J. Making Perceptual and Human Performance Data an Effective Resource for Designers. Proceedings of the NATO DRG Workshop (Panel 4). Royal College of Science, Shrivenham, England, 1984.

13. Hennessey, R.T. and McCauley, M.E. Proposal and Justification to Establish a Department of Defense Crew System Ergonomics Information Analysis Center. Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio. AAMRL-TR-87-XX, In Press.

14. Cody, W. and Rouse, W.B. Understanding and Aiding System Designers. Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, OH. AAMRL-TR-87-XX, In Press.

15. Reitman, W., Weischedel, R., Boff, K.R., Jones, M.E., and Martino, J. Automated Information Management Technology: A Technology Investment Strategy. Wright-Patterson AFB, Ohio. Air Force Aerospace Medical Research Laboratory, Report No. AFAMRL-TR-85-042, 1985.

ACKNOWLEDGMENTS

Many organizations have participated in and contributed to the IPID project. In particular, I am indebted to Professors Lloyd Kaufman (NYU) and Jim Thomas (UCLA) for their support of the Handbook phase of the project. Janet Lincoln, my co-editor on the Compendium, was a major source of ideas in the design of the Compendium concept. Edward Martin, the Project's Engineering Technical Advisor, made many important contributions to all phases of this effort. The staffs of the University of Dayton Research Institute and MacAulay-Brown Inc., Dayton, Ohio, played critical and leading roles in the development and production of the Handbook, Compendium and Design Short Course. I have personally enjoyed and profited from professional collaboration with Bill Rouse and members of the staff of Search Technology, Georgia, in refining the "Designer's Associate" concept. Overall, the project has significantly benefited from uncounted thousands of hours of professionally contributed time by many reviewers from the industrial, academic and government sectors. Finally, much of the success of the IPID project stems from the sustained encouragement, enlightened environment, and technical foresight provided by Charles Bates, Jr., Director of the Human Engineering Division and Thomas Furness, III, Chief of the Visual Display Systems Branch.

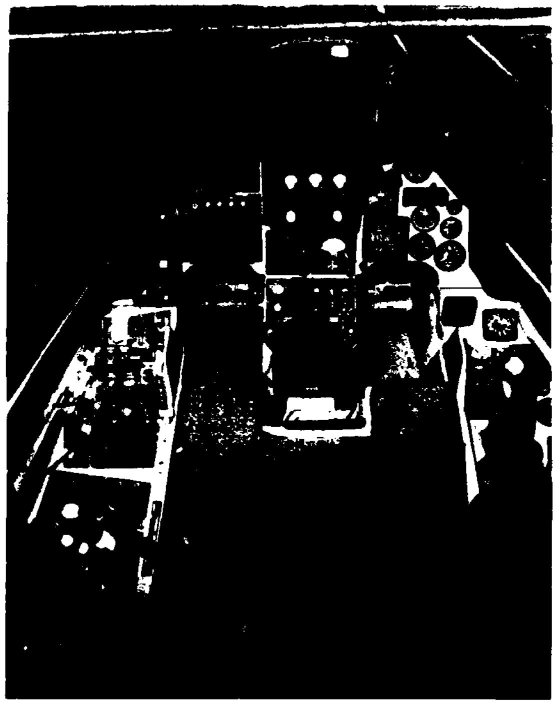


Figure 1

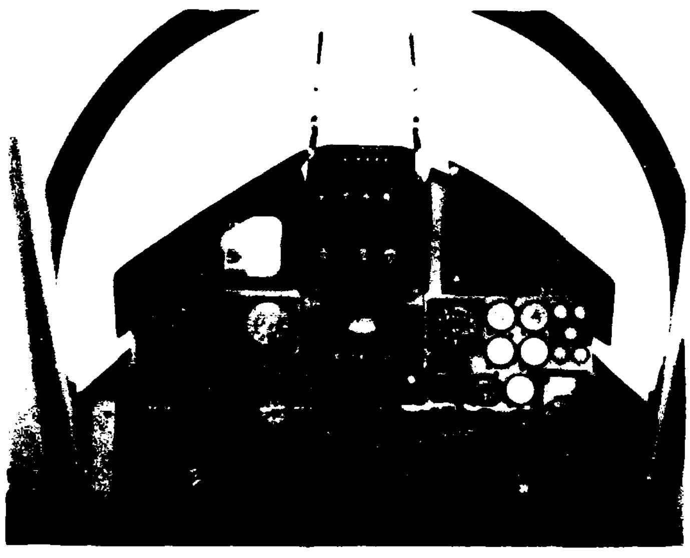


Figure 2

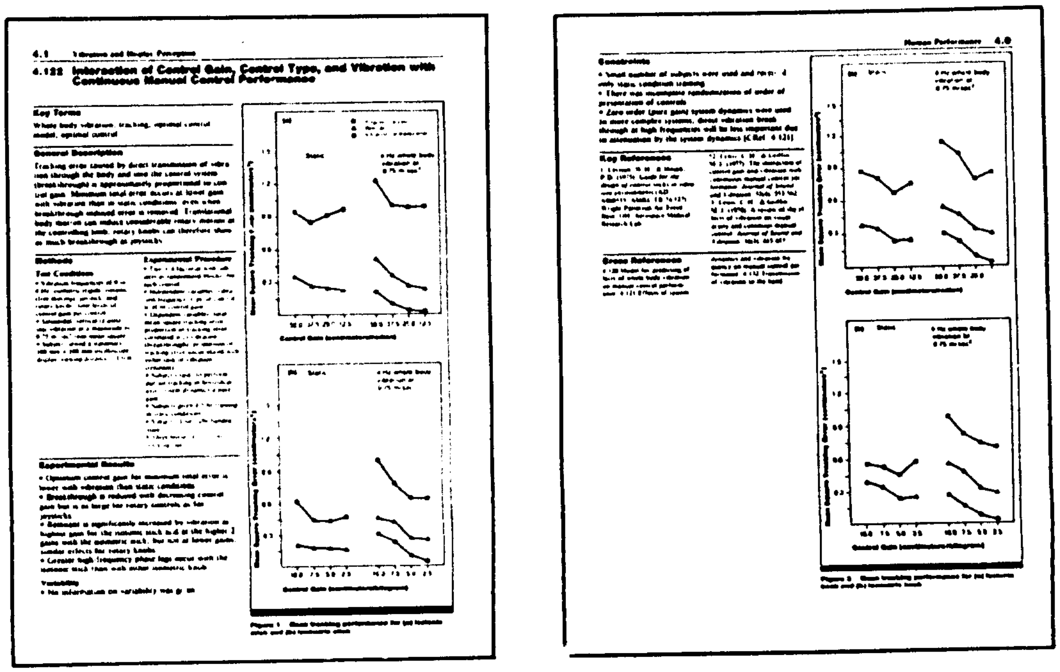


Figure 3 Sample compendium entry

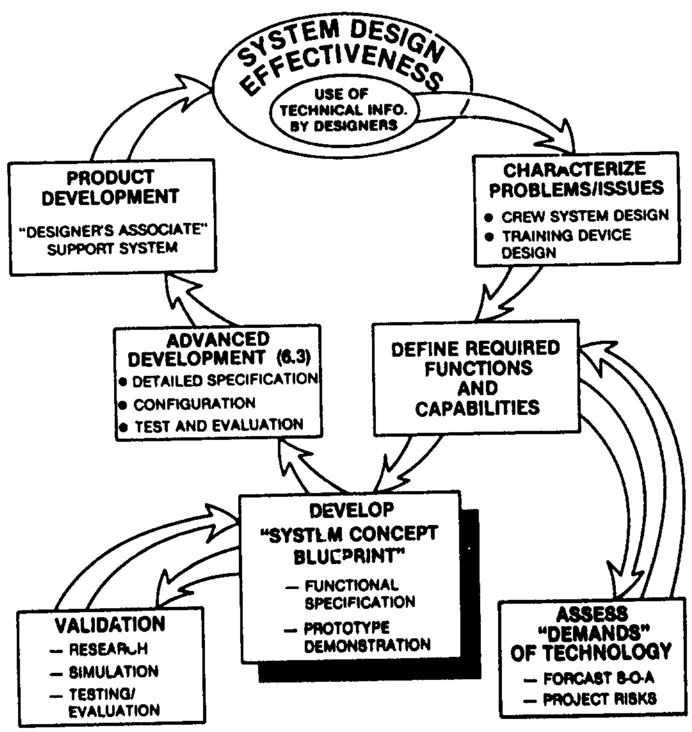


Figure 4

DESIGN FOR INTEROPERABILITY (INTERCHANGEABILITY)

By: George Konomos
AFWAL/AAAS-3
WRIGHT-PATTERSON AFB, OHIO 45433-6543

I. ABSTRACT:

Today the technology has reached the point that makes the communication among various subsystems easily realizable. On the other hand the complexity of these systems is such that their development is time and cost demanding. It is natural therefore to design our systems to communicate and work with each other and move away from the methods of the past.

Interoperability of the various elements used in a system is the design property which allows the intermixing of elements from various sources (manufacturers) without any impact on the performance of the system or the operational software.

One can arbitrarily distinguish at least three levels of interoperability.

a. Line Replaceable Unit. Although we are moving away from this architecture, the LRU approach requires a RADAR to use any interoperable computer rather than the one built by the RADAR manufacturer just for that purpose.

b. Line Replaceable Module. This is the new approach to avionics where a processor module is a 6" x 6" plug-in board with processing power many times higher than that of older LRUs. If it is in its infancy we certainly can design these modules to be interchangeable no matter who manufactures them.

c. VHSIC chip. This is the lowest level of meaningful interoperability. The various VHSIC developers have agreed to design their products "interoperable" with each other.

This paper deals with the second type. It is a case study of the two VHSIC computers we are now developing. We plan to use modules from two manufacturers to construct a small avionic system and then replace, both physically and electronically, any one of the modules with another made by the other manufacturer.

II. SOME IMPORTANT CONCEPTS:

It is difficult to imagine interoperable modules without really meaning interchangeable. In effect, we mean modules which can be made by various manufacturers and be able to be used indiscriminately in an avionics system. Some of these ideas were originated by the airlines. They demanded that any one can build a radio-communications "box" as long as it could be used in the airplane environment. That was a simple one. The only thing one had to worry about was the size, power and the radio frequencies it covered. Later, however, equipment became more complicated and had more than a couple parameters which interfaced with each other or the environment. The inertial navigation set, for example, needed some other input such as initial values. Eventually the airlines wrote specifications for purchasing this kind of equipment. Either because they were very smart and could distinguish the important from the non-important parameters or because they were not smart enough to specify manufacturing details, the result was that these specifications were short and simple. The military, on the other hand, was faced with the problem of developing special equipment which was not commercially available or had no use outside the military. The Heads Up Display (HUD) is such an example. Mainly because of these conditions, the military specifications became "development" specifications and gradually became more and more complicated. But interoperability does not deal with the manufacturing processes or the properties of the particular components used. Instead, it deals primarily with the visible by the user interfaces and performance characteristics which impact the rest of the system. We have attempted to have all this type of information in one document called "Functional and Interface Specification." The first one has been issued in cooperation with the two VHSIC MIL-STD-1750A module manufacturers and two users, system manufacturers.

In the case of the new "modular" electronics where, in fact, a "module" has hundreds of times the performance of an older "box" or LRU, the problem is very interesting. There are those who insist that the communications of such modules should be kept at very high level, thus simplifying the electrical and protocol problems of the connecting bus. Yet there are others who insist that this type of module can operate efficiently in a very tightly coupled system where each module is considered to be an extension of the other with extreme parallelism, reconfi- guration and fault tolerance. The second approach has been chosen for the first attempt of making "modular" electronics interoperable and interchangeable. In fact, the first such modules well defined at this time are the V1750A CPU modules provided with four channels of communications, namely; two PI-buses, one TM-bus and one IEEE-488 bus. The above mentioned Functional and Interface Specification reflects this definition.

An avionic system is comprised of a set of sensors and transmitters to evaluate and communicate with the aircraft's environment. These subsystems are connected to some type of data processors which manipulate information from and to sensors and transmitters. The processors, in turn, are connected to displays or other input/output devices for convenient man-machine interface. Although

it would be ideal if interoperability would apply to all the components of an avionic system, we will limit our discussion only to the processors. For this purpose, interoperability implies "physical or electronic replacement of one processor module with another without any impact on the Operational Flight Program (OFP)." It is important to understand that the only restriction we would have to apply to the OFP is that it must not include software timing code sequences or fault diagnostic routines dealing with hardware below the module level. These are not severe restrictions. Hardware timers are available on the modules and in an operational situation it is not necessary or important to isolate a fault to a particular capacitor. It is sufficient to isolate the fault to a module and then electronically remove it from the operation and perform its function in another interoperable module.

This type of interoperability implies the following conditions:

a. All modules must be dimensionally identical. This is necessary to support "physical" replacement.

b. All modules must have identical electrical interfaces. This also is necessary to support "physical replacement."

c. All modules must have identical Instruction Set Architecture (ISA). This is necessary to support the electronic replacement.

The first two requirements are easy to understand. After all we are used to the idea of replacing one voltmeter with another. It must fit in the existing mounting hole and must cover the same scales. But why do we need to have the same ISA in the modules? Doesn't that imply identical hardware? Doesn't Ada make the differences in the ISA invisible to the programmer?

The answer to the second question is "no." We have many implementations of the MIL-STD-1750A ISA. Each implementation uses very different hardware. Yet they all pass the software test we provide at WPAFB (MIL-STD-1750A validation).

The answer to the first question requires some assumptions. For example we must assume that our interoperable modules operate in a tactical fighter where "real time" is at a premium. "Real time reconfiguration" is necessary and must be accomplished at the minimum time possible. With such an assumption it becomes obvious that software programs must be stored in their "machine executable" form. We can not afford to store a program in its source form then compile it, link it and load it in a module while the fighter aircraft is in a high speed maneuver. In other words, software load modules must be stored in absolute machine code and ready for execution in any of the interoperable modules. This requirement of extreme efficiency in an operational environment demands identical ISAs in all processors within the same avionic system.

III. SOME CANDIDATE STANDARDS

In order to support this type of interoperability, the adoption of standards is necessary. In addition to MIL-STD-1750A ISA, the MIL-STD-1553B and the IEEE STD 488, the following should be considered as standards if interchangeabilty is to be taken seriously.

a) The Pi-Bus.

The Pi-bus version we have adopted is a 16 parallel bit, linear, multi-drop, synchronous bus which supports digital communications between up to 32 modules on a single backplane.

The bus uses a master-slave communications protocol which allows the bus master to read data from one slave or write data to any number of slaves in a single message sequence. Messages may be routed to particular modules using either logical or physical addressing. A number of independent messages may be transmitted during a bus master's tenure. The message formats provide a 32 bit virtual address range for each module.

The Pi-bus protocol specifies a set of bus state transitions which control the communication sequences and allow the bus to operate in a pipelined manner at the maximum clock rate allowed by the bus signal propagation delay. Master-slave handshaking is provided with a minimal performance penalty by operating the slave modules in synchronism with the master and using bus state look-ahead.

The bus provides a technique for temporarily suspending low priority block data transfers thus reducing bus acquisition latency for higher priority messages.

The bus mastership may be changed either by direct assignment or by priority arbitration. There are 128 logical levels of message priority and 32 levels of physical priority.

Extensive signal line and sequence error detection capability is incorporated into the bus definition. In addition, an optional single line error correction capability is specified.

b) <u>The TM-Bus</u>.

The Test and Maintenance (TM) bus is a simple, serial, linear, multidrop communications back-panel bus between a 'MASTER' module and up to 32 'SLAVE' modules residing in the same backplane. It operates with a clock of up to 6.25 MHZ. There are four signal types making the TM bus.

1) The clock, usually originating in a system clock module, separate than the master module,

2) TM-master data, a single uni-directional line used to transmit from the master the address, instruction data, scan data etc. to the slaves,

3) Slave data line for the slaves to transmit acknowledgments, data, and interrupts to the master, supporting wired-OR configuration, and

4) Control signal unidirectional from the master to slaves indicating DATA TRANSFER state when asserted or IDLE when released.

This bus is the companion of the PI-bus and uses most of the characteristics of it, including a derived clock.

Performance Characteristics	Protocol Characteristics
6.25 MHz clock (Typical)	8 reserved address bits
4 pin bus signal	32 module addresses (maximum)
Synchronous Operation	8 sub-addresses per module address
Two Data Lines	Multi-drop Configuration
SLAVE status register	Interrupt Capability

c) In addition, the DMA control and data structures as well as the XIO commands necessary for interfacing the MIL-STD-1750A processor and its memory with the PI and TM buses as well as the MIL-STD-1553B and the high speed data bus electronics should be considered for standardization.

VI. CONCLUSIONS.

The processor part of an avionic system is shown in figure 1. Various types of modules are interconnected to form a cluster and then these clusters are interconnected with the high speed data bus.

Based on our discussion, we have reached the conclusion that an avionic system can be built based on interoperable modules shown in figure 1. Each module is a processor with the left side I/O fixed. This includes two Pi-buses, one TM-bus and the IEEE-488 bus. We have made the right side variable to accommodate present and future external cluster interfaces, yet we keep the ISA part fixed.

To support this type of Avionic System we must accept the internal architecture of each module to be that shown in figure 2. This module requires for the left hand (BIU) interface five XIO commands and a DMA data structure. Similarly the right hand interface requires a number of XIO and identical DMA data structure.

Using exclusively this type of interoperable modules one can construct a high performance avionic system to accommodate:

a. <u>Fast real time reconfiguration</u> - Any module can be electronically added or subtracted from the system.

b. <u>Maximum fault tolerance</u> - Failed module can be electronically replaced by another, or at the worst case, its function can be moved to another module.

c. <u>Graceful degradation</u> - If remaining functioning modules can not perform all of the required functions, then functions of lesser importance can be eliminated. The functioning modules can be assigned to perform the more important functions.

d. <u>Simplified on line maintenance</u> - Modules have been designed for easy physical replacement on the aircraft. Because of their much smaller volume and weight, the "plug-in" feature, the visual fault indicators, and the extensive self-testing, there should be no need for highly skilled technicians to perform this task.

e. <u>Economic operation</u> - Avionics of various types of aircraft can be based exclusively on this type of modules thus requiring reduced stock piles of different kinds. In addition, the high demand of fewer types of modules will increase the buying quantities and encourage competition thus lowering the price.

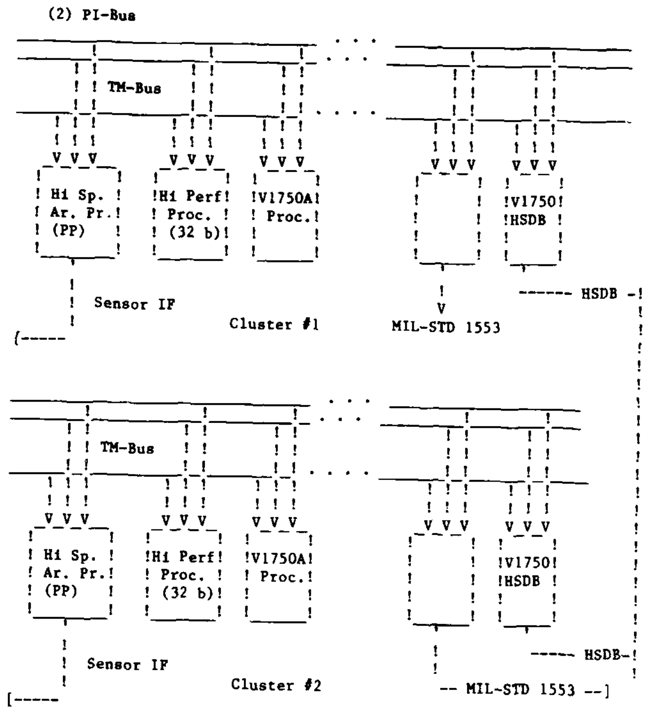
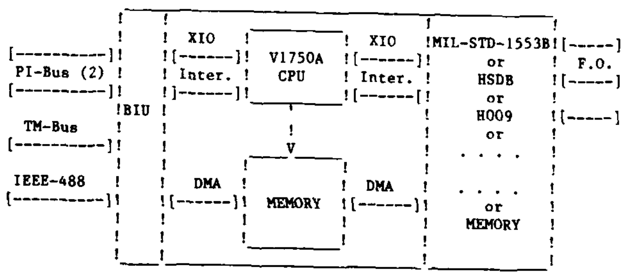


Fig. 1: Avionic System



Typical Interoperable Module

VII. REFERENCES

a) MIL-STD-1750A Notice 1 Sixteen-bit Computer Instruction Set Architecture (ISA)

b) VHSIC Phase 2 Interoperability Standards. PI-Bus Specification, Version 2.0 dated September 8, 1986 (By: IBM, Honeywell and TRW).

c) VHSIC Phase 2 Interoperability Standards. TM-Bus Specification, Version 1.2 dated August 31, 1986 (By: IBM, Honeywell and TRW).

d) IEEE-488 Standard Bus Specification. IEEE-488-1978 Standard, (as revised and supplemented in 1979 and 1980).

e) Module Functional and Interface Specification (FIS01R0 26 Nov 1986)
V1750A CPU (By: AFWAL/AAAS-3)

f) Module Functional and Interface Specification (FIS02R0 to be issued)
V1750A CPU/MIL-STD-1553B (By: AFWAL/AAAS-3)

g) Module Functional and Interface Specification (FIS03R0 to be issued)
V1750A CPU/High Speed Data Bus (By: AFWAL/AAAS-3)

h) Module Functional and Interface Specification (FIS04R0 to be issued)
V1750A CPU/Non Volatile Memory (By: AFWAL/AAAS-3)

THE ELECTROMAGNETIC THREAT TO FUTURE AVIONIC SYSTEMS

BRUNO AUDONE
AERITALIA Gruppo Equipaggiamenti
Caselle Torinese Torino (ITALY) 10072

SUMMARY

The electromagnetic threat to future aircraft is studied and evaluated on the basis of the evolution of the avionic systems. The high level of integration of these systems combined with the increased number of electromagnetic sources which may interfere with the performances of the overall weapon system create the need of reexamining the usual design and testing approach in order to reach an adequate level of aircraft hardening It is essential to design and test at system level rather than equipment level: this aim, obviously, is difficult to achieve mainly because of the lack of a well established methodology. The system design guidelines are discussed highlighting areas where basic research studies shall still be undertaken; similarly the system testing approach is discussed and a method of wide generality is described in detail. The Electromagnetic Compatibility (EMC) tests commonly devised for a system based upon the "black box" concept shall be formulated in a more general manner. The overall system hardening tests which are now carried out mainly on a qualitative basis shall be specified and performed on a quantitative one.

1. INTRODUCTION

In the design of present avionic systems developed according to the "black box" functional allocation concept the electromagnetic threat to avionic equipment/subsystems is specified in terms of different disciplines which take their names either from the source of the electromagnetic effect such as NEMP (Nuclear Electromagnetic Pulse), STATICS (electrostatic discharge) or from the equipment/subsystems affected such as HERO (Hazard of electromagnetic radiation to ordnance), HERF (Hazard of electromagnetic radiation to fuel), HERA (Hazard of electromagnetic radiation to avionics); moreover besides the threat of the unwanted internal or external hard electromagnetic environment there is the large spectrum of wanted interference/disturbances created for deception or jamming purposes: in this case one enters the broad domain of Electronic Warfare (EW).
By the replacement of the functional decomposition approach with the total system approach characterized by unclear definitive boundaries between subsystems the previous concepts based upon the selective discrimination of susceptibility effects is obviously substituted by the more general concept of the electromagnetic threat which will not affect a single black box but may interfere with the correct operations of the overall weapon systems.
The aircraft avionic system has followed a rapid evolution in the last years and is expected to have an even more rapid one in the future. Till twenty years ago the avionic system was an assemblage of equipments which performed certain tasks without any degree of integration whatsoever. Each equipment was a collection of black boxes each one connected to the power line with independent output data presentation. There was no electrical signal standardization apart from very specific cases such as the power line for example. Each equipment manufacturer was free of using its own electrical standard in terms of signal type and hardware (cables, connectors....). Since those years until now the avionic system has become an assemblage of subsystems related to a specific weapon system function (navigation, weapon delivery, flight control...) trying to achieve a higher level of integration. Because many of these subsystems consist of common elements it seems reasonable that the sharing of these elements among the various subsystems could result in a more reliable and efficient system design. Through the multifunctional use of the system elements it is possible to achieve a more highly integrated and cooperative avionic system with the remarkable advantages of reduced weight, volume and power consumption.
Additional advantages are:
- using a single set of sensors to satisfy different requirements (a single wideband antenna for communication and navigation)
- sharing processing resources to cope with various processing needs
- increasing the fault tolerance through crosscheck techniques

These advantages are achieved with increased communality among the various hardware parts of the system. For example in a subsystem oriented design the signal traffic between the inertial instrument and the navigation computer may not be visible beyond that subsystem.
Viceversa in a system oriented design with high level of integration where the inertial instrument is a sensor shared among the navigation, fire control and flight subsystems it is essential that the data traffic is compliant with a standard communication network. The existing bus standard is MIL-STD-1553B; it has evolved over the past ten years and represents the first significant step towards a highly integrated system.
It clearly appears that a high level of integration means a high levelof standardization: this fact has a heavy impact on the design and testing philosophy of the hardware components of the system.

2. THE ELECTROMAGNETIC THREAT

The electromagnetic environment where the aircraft is required to operate is becoming more and more crowded with powerful sources of electromagnetic energy which pose a severe system design requirement. On the other hand aircraft manufacturers are more and more using composite materials in the construction of structures with the benefit of reducing weight and cost with the disadvantage of reducing the values of shielding efficiency of the overall aircraft. The aircraft must be compatible with:

- the internal environment: it represents the electromagnetic fields generated by on board intentional and unintentional transmitters which may cover the full frequency spectrum up to 40GHz
- the external environment:it represents the electromagnetic fields generated by external intentional and unintentional sources. High electromagnetic fields are produced by broadcast radio transmitters and air to ground radars with power levels up to several megawatts; they may create safety critical situations in case of low level flight. Intentional transmitters include all those emitters whose main purpose consists in jamming communication and radar systems. There will be a large increase (both in power levels and quantity) of these types of emitters intended to intentionally interfere with electronically dependent systems.

Just recently it has been proposed to develop weapons capable of radiating the aircraft with extremely high level of energy.
These weapons use generators based upon new pulsed plasma magnetohydrodynamic (PPMHD) technology, which converts the chemical energy of an explosive cartridge directly into pulsed electrical energy.
A condition of critical external environment exists during formation flight when it is possible that an aircraft flies through the extremely high field strengths generated by the trasmitters of another aircraft. Lightning and Nuclear Electromagnetic Pulse (NEMP) are a further type of external environment which has also to be taken into account.
On the other hand integrated circuits are becoming available with higher operating speed, greater gate densities and lower power consumption: submicron devices operating at low voltage are being developed with the consequence that they will be more susceptible to interference.
The year 2000 projected RF profile at integrated circuit pins is shown in Fig. 1 [1].
These power levels are sufficient to create susceptibility effects to any unprotected solid state device over the frequency range 1MHz to 100GHz.

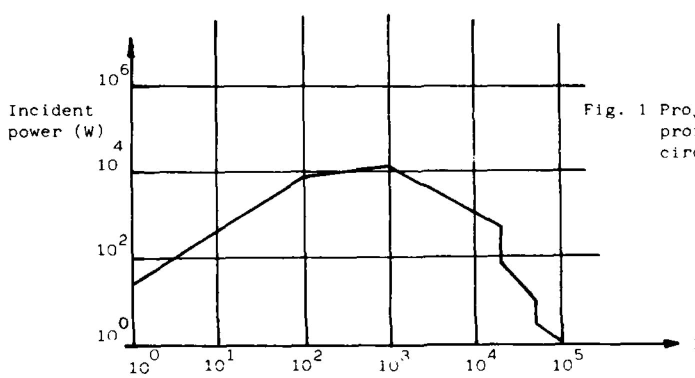


Fig. 1 Projected RF power profile at integrated circuit pins

One of the most common form of susceptibility consists in the rectification of the modulating signal of the unwanted incident radiation: a pulse modulated out of band signal may cause the unwanted trigger of the affected device. One can define the rectification efficiency R as the ratio of the rectified DC voltage at the device output to the RF power level incident at the input (for an unmodulated signal). In case of bipolar transistors:

$$R = K_B \frac{1}{\sqrt{\omega}} \frac{1}{p} \left[\frac{V}{W}\right]$$

where

K_B is a constant depending on the input resistance, capacitance and emitter area

p is the emitter perimeter

$w = 2\pi f$ is the operating frequency

In case of MOS devices

$$R = K_M \frac{1}{\omega^2} \frac{1}{C_g} \left[\frac{V}{W}\right]$$

K_M is a constant depending on the parasitic resistance, the transconductance

C_g is the input gate capacitance

From the previous equations it appears that the rectification efficiency versus frequency of bipolar transistors is higher than the one of MOS devices. In bipolar transistors reducing the size of the component will decrease the device perimeter with the obvious increment of R. In case of logic circuits interference effects, while are not particularly harmful to signal processing systems, may have catastrophic effects in decision maker systems such as signal sorters in EW equipments where pure logic operations are carried out. The susceptibility of logic circuits depends upon the noise immunity level (NIL) which is defined as the least DC noise margin that exists between gate output and input voltages in conditions of "0" and "1". At present operating with TTL levels typical values of NIL's are 0.4V; but in future circuits with ECL technology it is expected to operate with values of NIL's of about 0.1V with the obvious increase probability of interference.

3. SYSTEM LEVEL DESIGN

With the total system approach where the subsystems are provided by different suppliers it becomes essential to establish suitable standards as for example it has already been done for the data bus. A standard to satisfy the electromagnetic threat hardening requirement shall be established at the beginning of a new project. These standard guidelines shall deal with all those aspects of the equipment/subsystem design which are related to design aspects usually treated by the EMC discipline. They are:

- grounding
- bonding
- filtering
- shielding
- connector and wiring
- electrical interfaces

The guidelines shall not be a general collection of common sense design rules (as it generally happens); but shall clearly identify the components, the values of electrical parameters which can guarantee an adequate margin of safety against the foreseen electromagnetic threat. This task is obviously difficult because it means that more responsibility shall be taken by the aircraft manufacturer or by the engineering team who has the design responsibility task.

The grounding concept and the electrical interfaces represent keypoints in the definition of a system. Therefore they will be discussed in detail.

The general lay-out of an electronic/electrical equipment can be sketched as shown in Fig. 2.

In this diagram different types of grounds are indicated: 1 and 1' are the DC and AC power grounds respectively, 2 is the signal reference ground (SRG), 2' is the virtual signal reference ground of balanced receivers which may be isolated from 2, 3 is the equipment structure ground.

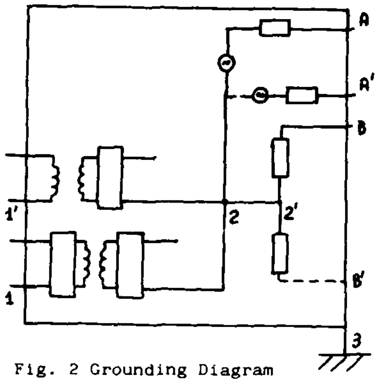


Fig. 2 Grounding Diagram

A unbalanced output
AA' balanced output
B unbalanced input
BB' balanced input

Separate returns for AC and DC power grounds are commonly implemented for most equipments. The area of major uncertainty is represented by the manner of treating the SRG 2: it may isolated or connected to the equipment structure ground.
In the interconnection of the equipments of a system there are four fundamental ways of doing it [Fig. 3]. One method is to isolate the SRG from the equipment case at the source and at the load providing the necessary shielding and filtering to avoid unwanted coupling via other means (Fig. 3a). This is the Floating Grouding (FG) system.
It suffers from many disadvantages: the main one is the static charge build up on the equipment case with the possible consequence of spark hazard to the internal circuitry. Another grounding scheme is the Single Point Grounding (SPG) system where the SRG of all the equipments is connected to a Central Signal Point Ground (Fig. 3b). In this manner there is no problem of common mode interference. This type of ground system requires a very large number of conductors and therefore is not feasible mainly because of weight implications. The main drawbacks of the SPG grounding scheme are:

- the difficulty of implementing the isolation between the SRG and the case within RF equipments in conjunction with the increase of stray capacitance coupling;
- the reduction of the shielding effectiveness of equipment cases due to the grounding wire penetrating the equipment structure and therefore violating the metallic barrier. This is particularly detrimental for systems where the EMP protection is required.

The third grounding scheme is the Multiple Point Grounding (MPG) system where the SRG is directly connected to the equipment structure ground (Fig. 3c). The common mode noise represents the greatest problem; the reduction of this type of interference is obtained by striving for a zero impedance reference plane. If a truly zero impedance ground plane could be built, it could be used as the return path for all currents (power signal and RF). Unfortunately the aircraft fuselage structure is far away from an ideal zero impedance plane. The advantages of the MPG system are:

- to make the RF equipment design easier because within the equipment the case offers a ground plane better than any wire and to avoid complex decoupling systems;
- to improve the shielding effectiveness of the equipment because the metallic barrier of the case is not violated;
- to eliminate spurious capacitive coupling.

The last grounding scheme is the Distributed Single Point Grounding (DSPG) system where the SRG is connected to the equipment structure ground but the input and output interfaces are differential balanced circuits with high levels of Common Mode Rejection Ratio (CMRR) (Fig. 3d). This grounding method is probably the best one because it combines the advantes of SPG and MPG systems. In order to have a high CMRR it is necessary to have a true balanced system at the source, at the load and along the connecting line: the impedances along the two wires of the transmission path shall be perfectly equal.

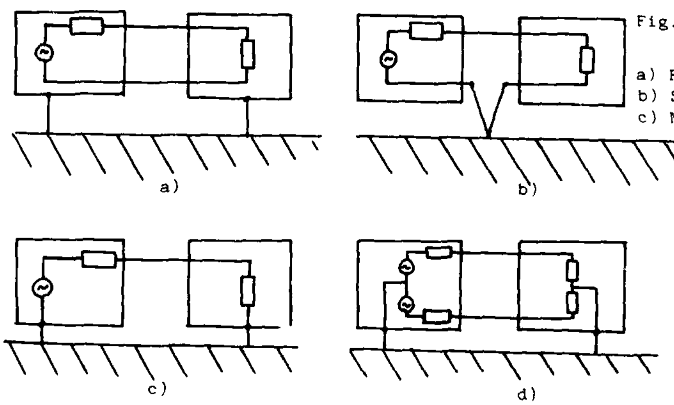


Fig. 3 Grounding Diagram

a) Floating G (FG)
b) Single Point G(SPG)
c) Multiple Point G(MPG)
d) Distributed Single point G (DSPG)

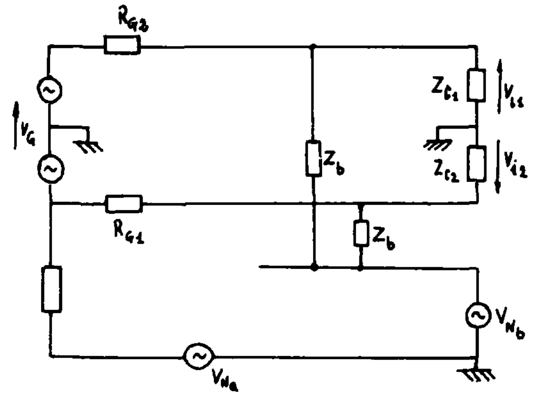


Fig. 4 Schematic diagram of a balanced circuit

In Fig. 4 the schematic diagram of a balanced circuit is shown. The following parameters can be defined

$$\Delta R_G = R_{G1} - R_{G2}\ ;\ R_G = \frac{R_{G1}+R_{G2}}{2}\ ,\ K = \frac{\Delta R_G}{R_G}$$

$$\Delta Z_c = Z_{c1} - Z_{c2}\ ;\ Z_c = \frac{Z_{c1}+Z_{c2}}{2}\ ,\ \lambda = \frac{\Delta Z_c}{Z_c}$$

It is possible to calculate [3] the transfer functions of the wanted signal, common mode noise V_{Na} and the capacitively coupled noise V_{Nb}

$$T_G = \frac{V_{id}}{V_G} \simeq \frac{Z_c}{R_G+Z_c}$$

$$T_{Sa} = \frac{V_{id}}{V_{Na}} \simeq \frac{Z_c}{R_G+Z_c}\,\frac{R_G}{R_G+Z_c}\,(\lambda-K)$$

$$T_{Sb} = \frac{V_{id}}{V_{Nb}} \simeq H\,\frac{Z_p}{Z_b}$$

where:

$$V_{id} = V_{i1} - V_{i2}\ ;\ Z_p = \frac{R_G \cdot Z_c}{R_G+Z_c}\ ;\ H = \frac{Z_c}{R_G+Z_c}\,K + \frac{R_G}{R_G+Z_c}\,\lambda$$

The following ratios are meaningful

$$A = \frac{T_G}{T_{Sa}} \approx \left(\frac{Z_c}{R_G}+1\right)\frac{1}{\lambda - K}$$

$$B = \frac{T_G}{T_{Sb}} \approx \frac{Z_b}{R_G}\,\frac{1}{H}$$

The output voltage of the differential receiver is given by

$$V_o = A_d V_{id} + A_c V_{ic}$$

where $V_{ic} = \dfrac{V_{i1}+V_{i2}}{2}$

where A_d is the differential mode amplification and A_c is the common mode amplification. In differential receiver, it is common practice to introduce the common mode ratio CMR defined as $CMR = \dfrac{A_d}{A_c}$

By suitable substitution it is easy to calculate the output voltage in presence of common mode noise V_{Na}

$$V_o \approx T_G A_d \left[V_G + \left(\frac{1}{A} + \frac{1}{CMR}\right)V_{Na}\right]$$

and in presence of the capacitively coupled noise V_{Nb}

$$V_o \approx T_G A_d \left[V_G + \left(\frac{1}{B} + \frac{1}{CMR}\,\frac{R_G}{Z_b}\right)V_{Nb}\right]$$

From the previous equations some comments are in order:

- the CMR of the differential amplifier alone is not sufficient to guarantee a good rejection ratio if the overall differential line is not properly balanced
- the degree of balance depends on the line and on the transmitter where it is required to have either the same unbalance as at the receiver or a very low resistance
- the capacitively coupled noise does not only depend upon CMR but is also related to $\dfrac{R_G}{Z_b}$.

The balanced transmission line is a typical example of a system oriented problem. It is useless to require a very high CMR at the input terminals of an equipment if at the same time adequate precautions are not taken at the transmitter end of the line.
As it has been stated in the previous paragraph the DSPG system is mainly based upon suitable interfaces between the equipments: the basic components are the isolation amplifier and the differential amplifier. The isolation amplifier is a device which provides ohmic isolation between the input and the output . It is particularly suitable for applications requiring accurate measurement of low frequency voltage in the presence of high common mode voltage (thousands of volts).
The differential amplifier is a device which responds only to the difference voltage between inputs and produces no output for a common mode voltage.
Both components are characterised in terms of the Common Mode Rejection Ratio (CMRR).
Operational amplifiers are susceptible to RF energy conducted into either of the input terminals.
When stimulated in this manner the interference effect is an offset voltage at the particular input terminal entered by the RF: this offset voltage may be either a DC level or an undesired low frequency response due to demodulation effects.
The magnitude of the offset voltage depends on such factors as the power level, frequency equivalent RF source impedance and the op. amplifier input circuit.
Demodulation RFI effects are greater in operational amplifiers with bipolar input transistors (741 and LM10) than they are in operational amplifiers with MOSFET input transistors (CA081) and with JFET input transistors (LF355).
At RF frequencies above 10 MHz demodulation RFI effects in the 741 op amplifier are significantly greater than in the LM10 op. amplifier. This is possibly a result of the cutoff frequency of the npn bipolar input transistors in the 741 op. amplifier being higher than the cutoff frequency of the less conventional (pnp substrate) bipolar input transistors in the LM10 op. amplifier.

4. SYSTEM LEVEL TESTING

Till now the major part of the testing activity related to the verification of the equipment hardening against external electromagnetic interference has been performed within the domain of the EMC susceptibility tests. These tests are carried out according to one of the numerous EMC specifications existing in the world which are all more or less derived from MIL-STD-461/462A/B. These tests are divided in two broad categories: radiated susceptibility and conducted susceptibility. Different types of signals (transient and RF) are injected by conduction on power lines and signal lines or by radiation on the overall equipment. During the design of a new aircraft these tests are specified by the main contractor to the equipment/subsystem suppliers. Even if they are accurate in the definition of the test set up and in the description of the test procedure, they have two main drawbacks:

- the type of testing implies the design at equipment level and is not related to a system oriented design
- the susceptibility criteria are not clearly specified and in any case imply malfunctions at equipment level

The former point is particularly misleading because the system oriented design, by its very nature, operates in terms of standards (data bus, connector, cable, shielding, signal format); therefore when the supplier is asked to carry out susceptibility tests it is not always clear whether he is testing the quality of its equipment or the quality of the standard he was requested to use. Additionally the test (especially the radiated one) may not be fully representative because the lay out of the overall subsystem does not represent the actual situation in terms of equipment location, cable length and routing. In a highly integrated system it is difficult to have access to parameters to be monitored without affecting the system under test. In view of all these facts it is essential to develop test methods suitable to match the system oriented design and capable of performing quantitative measurements. Simple qualitative checks intended to discover gross interference/susceptibility situations on the basis of functional interactions are no longer adequate to clear complex systems: there is a well defined need of measuring a quantitative level of safety before equipment malfunction occurs.
In some cases, because of its nature or of the internal noise, the parameter of the equipment under test changes randomly around an average level without the presence of the interference source; therefore it may be difficult to find out whether the parameter is affected when the interference source is activated. This problem can be overcome by means of a statistical approach which can also be useful to establish the safety margin. The basic parameters of a random variable x which specify its central tendency and dispersion are the mean value $\hat{\mu}_x$ and the variance $\hat{\sigma}_x^2$ defined as

$$\hat{\mu}_x = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \hat{\sigma}_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \hat{\mu}_x\right)^2$$

These estimators are unbiased, efficient and consistent for the actual mean value μx and actual variance σ_x^2 of a random variable; however they result only in point estimates for a parameter of interest. A more meaningful procedure for estimating parameters of random variables involves the estimation of an interval, as opposed to a single point value, which includes the parameter being estimated with a certain degree of uncertainty. Such an interval can be determined if the sampling distribution of the estimator is known. For the case of the variance σ_x^2 based upon a sample variance $\hat{\sigma}_x^2$ computed from N samples a confidence interval can be calculated as

$$\frac{n\,\hat{\sigma}_x^2}{\chi^2_{n;\frac{\alpha}{2}}} \leqslant \sigma_x^2 < \frac{n\,\hat{\sigma}_x^2}{\chi^2_{n;1-\frac{\alpha}{2}}}$$

with

$$\int_{\chi^2_{n;\frac{\alpha}{2}}}^{\infty} Q\left[\chi_n^2\right] d\chi_n^2 = \mathrm{Prob}\left[\chi_n^2 > \chi^2_{n;\frac{\alpha}{2}}\right] = \frac{\alpha}{2}$$

where

$Q\left[\chi_n^2\right]$ is the Chi -square distribution function ; $n = N-1$

The degree of trust associated with the confidence statement is $1-\alpha$ and is called "confidence coefficient". Furthermore if σ_x^2 is unknown, a confidence interval can still be established for the mean value μx based upon the sample values $\hat{\mu} x$ and $\hat{\sigma} x$ as follows

$$\hat{\mu}_x - \frac{\hat{\sigma}_x\, t_{n;\frac{\alpha}{2}}}{\sqrt{N}} \leqslant \mu_x < \hat{\mu}_x + \frac{\hat{\sigma}_x\, t_{n;\frac{\alpha}{2}}}{\sqrt{N}}$$

with

$$\int_{t_{n;\frac{\alpha}{2}}}^{\infty} P\left[t_n\right] dt_n = \mathrm{Prob}\left[t_n > t_{n;\frac{\alpha}{2}}\right] = \frac{\alpha}{2} \qquad n = N-1$$

where $P\left[t_n\right]$ is the Student Distribution function

The degree of trust associated with the condifence statement is $1-\alpha$ and is called "confidence coefficient".

The practical implementation of the previous theory is described in the following. Suppose the sensors (R. Altimeter, Radar ...) are connected to the data bus through coupler boxes as shown in Fig.5. A Fiber Optic Transmitter is installed within the aircraft structure in the nearest possible position to its coupler box. The fiber optic link connects through the aircraft fuselage the transmitter to the receiver located in the remote position where the bus Analyzer receives the parameters to be examined. The bus analyzer and the RF environmental generators are driven by the computer which establishes the sequences of the testing procedure which is divided in these steps:

a) evaluation of sample mean, sample variance and confidence intervals for the parameters under test on the basis of the subsystem performance specification
b) evaluation of sample mean, sample variance and confidence intervals for the parameters under test with the on board subsystems in fully operative conditions
c) activation of the RF environmental generators and measurement of sample mean, sample variance and confidence intervals of the parameters under test at different frequencies, amplitude and modulations of the RF environmental generators .

In step a) one determines the sample mean μo and its relevant confidence interval (Lo, Uo), the sample variance σ_o^2 and its relevant confidence interval (Lo', Uo'). These represent the reference conditions. In step b) one determines the sample mean $\hat{\mu}_N$ and its relevant confidence interval (L_1, U_1) the sample variance $\hat{\sigma}_N^2$ and its relevant confidence interval (L_1', U_1'). An interference margin at system level may be defined as follows.

$$IM_{sys} = 20 \log \frac{L_o}{L_1} \qquad \hat{\mu}_N \leqslant \mu_o \qquad ; \qquad IM_{sys} = 20 \log \frac{U_1}{U_o} \qquad \hat{\mu}_N \geqslant \mu_o$$

$$IM_{sys} = 20 \log \frac{L_o'}{L_1'} \qquad \hat{\sigma}_N^2 \leqslant \sigma_o \qquad ; \qquad IM_{sys} = 20 \log \frac{U_1'}{U_o'} \qquad \hat{\sigma}_N^2 \geqslant \sigma_o$$

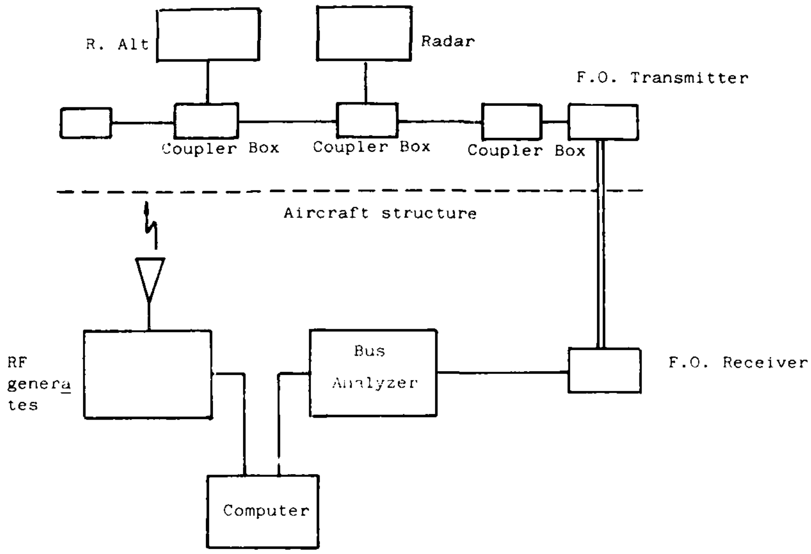


Fig. 5 Test Set up for system level test

An interference situation ($IM_{SYS} > 0$) in the fully operative conditions is a clear symptom of an integration malfunction to be solved at system level. It is necessary to solve the problem before proceeding to step c). Viceversa when $IM_{SYS} < 0$ the RF environmental generators can be activated. In step c) one determines the sample mean $\hat{\mu}_{N+I}$ and its relevant confidence interval (L_2, U_2), the sample variance $\hat{\sigma}^2_{N+I}$ and its relevant confidence interval (L_2', U_2'). An effective interference margin can be defined as follows: (Fig. 6).

$$IM_{EFF} = 20 \log \frac{L_0}{L_2} \qquad \hat{\mu}_{N+I} \leq \mu_0$$

$$IM_{EFF} = 20 \log \frac{L_0'}{L_2'} \qquad \hat{\sigma}^2_{N+I} \leq \sigma_0$$

$$IM_{EFF} = 20 \log \frac{U_2}{U_0} \qquad \hat{\mu}_{N+I} \geq \mu_0$$

$$IM_{EFF} = 20 \log \frac{U_2'}{U_0'} \qquad \hat{\sigma}^2_{N+I} \geq \sigma_0^2$$

An interferent situation exists when $IM_{EFF} > 0$.

The evaluation of the confidence interval is easily performed for a fixed level signal whose level and accuracy are known. This signal may be represented as follows:

$$a(t) = K \pm \Delta K$$

where:

K = nominal signal level
Δ K = maximum deviation
The mean value confidence interval (Lo, Uo) may be assumed ($K - \Delta K$, $K + \Delta K$) with a confidence interval equal to 100%. The sample mean is K.
The variance confidence interval (Lo', Uo') may be assumed (0 , ΔK^2) with a confidence interval equal to 100%. The sample variance is $\frac{\Delta K^2}{2}$.

These measurements can be performed at different frequencies, amplitudes and modulations of the interference source. It is also possible to evaluate a correlation coefficient between the parameter under test and the parameter of the interference source in order to establish possible relationships which may be useful in the determination of the

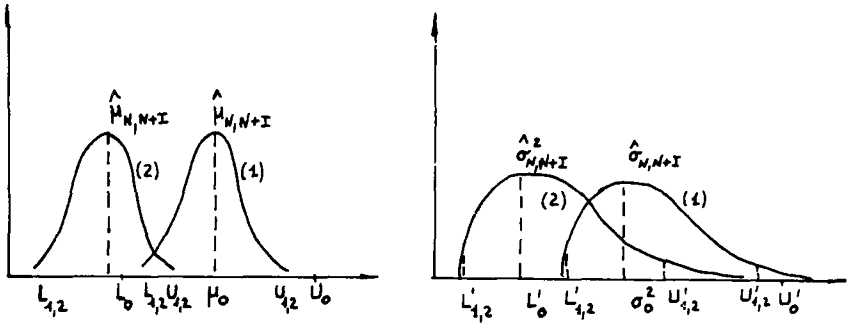


Fig. 6 Examples of sampling distribution functions

1) non interferent situation ($L_{1,2} > Lo$, $U_{1,2} < Uo$) and ($L'_{1,2} > L'_o$, $U'_{1,2} < U'_o$)

2) interferent situation ($L_{1,2} < Lo$ and $L'_{1,2} < L'_o$)

source of the problem.
These tests shall be performed at rig level and at aircraft level. There may be some difficulties in performing radiated tests at rig level because the rig is generally located within a laboratory. In this case it may be useful to perform the so called bulk current injection tests (BCI). The BCI testing procedure involves current injection into the electronic component under test by toroidal transformer excitation of the wiring harness. The bulk current injected into the component is monitored via a toroidal probe Fig. 7. The advantage of this test method consists in the possibility of avoiding radiating the rig with the antenna and localizing the interference on a selected wiring harness.
There are some disadvantages which are:

- the need of a multiple injection probe system in case the subsystem has many wiring looms which shall be interfered at the same time in order to avoid unrealistic upsetting of the subsystem
- the injected signal depends on the position of the probe along the cable ; it is necessary to perform the test with the probe located in different positions
- the frequency range is limited up to 400 MHz

Radiated susceptibility tests [5] can also be performed by radiatingthe system under test by means of TEM cell which consists of a section of rectangular coaxial transmission line with tapered sections at both ends. The taper acts as a transition to match the line to a 50 Ohm coaxial line at the two ports of the cell.
Unfortunately TEM cells have a limitation on operating frequency because the TEM mode exists only at low frequency. Moreover since the polarization of the field is fixed the radiated susceptibility test requires physical rotation of the equipment under test.
Another technique which does not require the rotation of the equipment under test uses reverberating enclosures [6] to generate an average homogeneous and isotropic field within a metal chamber. The homogeneous and isotropic field is obtained by rotating a tuner whose purpose consists in perturbing the possible modes existing within the cavity.
Reverberating chamber technique is good for applications at high frequencies; therefore they may be used as supplementary tools to TEM cells.
Susceptibility tests can be performed in many ways; in all cases the aim is to simulate the environmental conditions where the equipment under test shall operate.

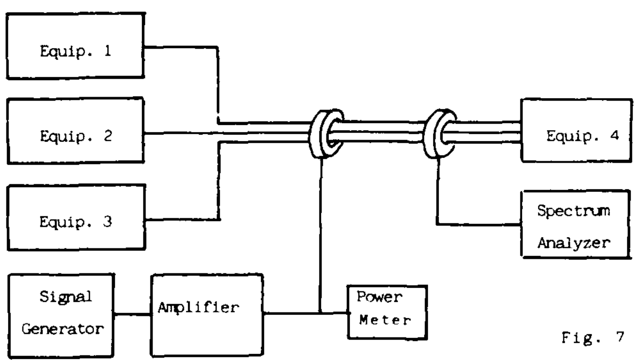


Fig. 7 - Bulk current injection test method

5. CONCLUSION

The electromagnetic threat will be the challenge to future avionic systems. Therefore suitable design rules and testing techniques shall be established in order to built the aircraft with an acceptable margin of safety: the aim can be met by designing and testing the aircraft at system level . Careful selection of electrical standards during the design phase combined with the system level testing approach represent the key points to be successful in this difficult task.

6. REFERENCES

|1| H.W. Denny "Projected Susceptibilities of VHSIC VLSIC Devices to the year 2000 Electromagnetic Environment" 1986 IEEE Symposium on EMC

|2| Y. H. Sutu and J.J. Whalen "Statistics for demodulation RFI in Operational Ampl. fiers" IEEE Symposium on EMC Zurich 1983

|3| Gottwald A:"About the suppression of spurious signals by balancing the transmission circuitry"Proc. of the 3th Int. Symp. on EMC Wroclaw (1976)

|4| B. Audone, R. Cazzola and G. Barale "Statistical evaluation of the EMC safety margin at system level "IEEE Symposium on EMC Montreux 1984

|5| M.L. Crawford and J.L. Workman "Using a TEM cell for EMC measurements of electronic equipment" TN1013 NBS Boulder 1981

|6| P. Corona et alias "Performance and analysis of a reverberating enclosure with variable geometry" IEEE Transactions on EMC Vol. EMC-22 FeB 1980.

THE INTEGRATION, CHARACTERISATION AND TRIALLING OF A MODERN COMPLEX AIRBORNE RADAR

Author: Roger Roy Hogben. C.Eng.
Senior Systems Consultant.

Co-author Frank N. Morphet. B.Eng.
Chief Engineer

Organisation: THORN EMI Electronics Limited,
Radar Division,
Radar House 2,
Dawley Road
Hayes,
Middlesex. UB3 1HN
U.K.

SUMMARY

The inception of a radar system is usually marked by the building of a complete prototype or pre-production equipment. Systems engineers then have to prepare for and implement a work programme to integrate and characterise the equipment at the works as comprehensively as possible before installation for flight evaluation in a suitable trials aircraft.

This paper sets out to examine the process of commissioning, testing and trialling of a complex airborne radar and attempts to show how vital an informed and methodical approach is to achieve success. Indeed neglect or parsimonious treatment of these activities can lead not to savings but to chaos or at best, prolonged iteration with substantial overspending and lengthening delays.

1. INTRODUCTION

The author has been engaged in systems engineering activities in this field with the Radar Division of THORN EMI Electronics over the past fifteen years. The principal equipments involved have been prototype and development models of the Searchwater maritime reconnaissance (MR) radar, the maritime AEW derivative of Searchwater, the one-off installation of Searchwater MR in a US Navy P3B and the current low cost AEW Skymaster radar.

Searchwater in either form is a frequency agile pulse compression radar featuring extensive signal processing, and sizeable computer content. Skymaster, which also features coherent operation for pulse doppler AEW, is a modernised lightweight derivative largely configured with multi-processors.

Searchwater MR is in squadron service with the Royal Air Force in BAe Nimrod MR MKII aircraft. It should perhaps be noted that this variant of Nimrod and its radar have fulfilled tneir role very successfully over several years and have no connection with the aborted AEW MKIII project other than a common airframe.

Searchwater AEW, which originally evolved as an emergency adaption for the Falklands conflict, is in squadron service with the Royal Navy in Westland Sea King helicopters. It is also being supplied to the Spanish Navy for use in Sikorsky SH3D helicopters.

Skymaster is currently undergoing flight evaluation in a Pilatus Britten-Norman Defender aircraft.

2. PREPARATION

It is important that the system engineer(s) who is to see the prototype equipment through to the conclusion of flight trial evaluation should be involved at the early stages of project development and design. Several benefits should accrue from early involvement; for instance, in the period since the previous major project of a similar kind, the world of technology will have moved on prompting a quest for updating even the most broadly informed individual's knowledge. Similarly system techniques known but outside the individual's direct experience may require time for inquiry and familiarisation. In the last resort it is the practical systems engineer who will be expected, in the rough and tumble world of extracting good performance from new equipment, to provide much of the inter-discipline liaison and interpretation between the often conflicting interests of differing specialists. Nowhere in this divergence of interests more likely to occur than between those with basically hardware orientated allegiance and those with software responsibilities.

Benefit can also be realised at this stage from past related experience in such matters as interfacing policy, ergonomics and previous dealings with equipment operators in the armed services.

Nevertheless, despite this it is important that the systems evaluation engineer does not become too involved or tied down in the detail engineering: that is for others. The first objective must be to come to terms with the physical principles upon which radar performance will hinge. The engineer should feel that his or her thinking has got "inside" the system. It is not essential to understand in detail, circuitry, machine code or the more esoteric performance mathematics but it is crucial to appreciate the radar system and its arena meaning the customer's requirement, the fundamental physics both within the radar and in free space, the computing, the hardware and the system role within the total aircraft avionics.

The system may now be examined with more confidence and dissected into its component parts, for example the r.f. section, analogue processing, gain control, thresholding, digital processing, data computation, displays etc. It is often helpful to draw or redraw functional block schematics to improve understanding at this stage. This type of activity assists in identifying the key points for monitoring; for example detector outputs, gain and threshold settings. As systems become more complex, compact and interwoven with an associated dissolution of unit or "black box" composition identification of clear key interfaces for monitoring becomes more difficult. The value and utility of straightforward monitor or test points cannot be overstated. In cases where vital data is neither readily available nor easily decypherable, modifications or special provision should be suggested. These comments apply both to the circuitry and the software data areas.

At this same time vital performance points and test objectives can be identified. Particular attention should be paid to those tests whose results can stand as a control reference when the programme moves on to the flight trials stage. Also likely to be very useful for reference purposes throughout testing and trials are reversionary type control options such as those that remove automatic operation or open large feedback loops. Typical examples are manual rather than automatic receiver gain control and manual rather than CFAR setting of thresholds. It may be prudent to introduce options of this type even where no long term operational requirement exists.

Where reviewing the system key performance features those associated with navigation inputs to the radar merit close scrutiny. Motion compensation errors from aircraft navigation sensors have been known to severely impair overall radar performance once airborne. Due to the different disciplines involved, it has not been unknown for the Inertial Navigation System (INS) contractor to have a genuine but very different interpretation of a key specification clause to that of the radar designers.

A system test rig or area has to be prepared in readiness for the arrival of the main equipment. Basic requirements include ample space for an accessible layout, power, temperature conditioning together with support infrastructure such as cable gantries and benches. In addition commercial items such as D.C. power supplies, keyboards, printers and capital test equipment have to be ordered or obtained.

Staffing the whole operation through to flight trialling has also to be considered. The author's experience is that progress is best sustained by a small nucleus of full time staff with support from project specialists as and when appropriate. The value of a good artisan, particularly a prototype wireman, to implement permissible modifications and construct special to type test apparatus ought not to be overlooked. It is sound policy to introduce trials or field engineers to the system concept at this stage prior to participation in the later system characterisation measurements.

3. INTEGRATION AND COMMISSIONING

A schedule for these activities should be drawn up to shown in some detail the sequence, nature, effort and anticipated time for each step. The procedure outlined may be smartly, even sharply timed but it should always be methodical and include time for regular written reports. The schedule may well be widely agreed at the outset but too often projects run late due to delays for example, in design or card manufacture increasing pressure on the system programme. System integration, characterisation and trialling being the last in line of all development activities there is a very real danger of expedient policies being suggested. The argument may even be advanced that the system merely requires assembly, switching on and all will be fine. Unfortunately with complex equipment this is never so, for umpteen sundry reasons. Total acquiescence to this type of policy therefore is very likely to result in disorganisation and further delay. In the event therefore schedules may have to be reviewed carefully but not severely compromised.

Built-In-Test-Equipment (BITE), once itself commissioned, is a useful aid to system integration and problem diagnosis but the rig team ought also be watchful for other ways in which the radar can in effect check itself, without resort to expensive test equipment. For instance, where alternative modes or facilities exist each may be cross-checked against the other. It is also helpful to assess and establish the system quiescent conditions in noise limited conditions. These include receiver gain demand, mean detector output, threshold setting and overall false alarm rates. A confident knowledge of these simple parameters allows a rapid daily or pre-measurement check to be made; for random component failure may occur at any time to confuse the situation and perhaps invalidate lengthy measurement results.

Self test, formal or improvised, is of course never enough to chart performance sufficiently to ensure a confident transfer of the equipment to an aircraft. Simulators, sophisticated and simple, are also required to provide amongst other things interface checks, synthetic target generation and synthetic aircraft motion (linear and manoeuvre).

The rig will also require appropriate video and data recording together with corresponding replay facilities. Some of this will be common with the planned aircraft instrumentation. It may be appropriate to modify the rig signal processing to absorb and reprocess in-flight video to be recorded later from a tap at the corresponding point in the signal chain. "Real" data of this form, even though inherently frozen beyond adjustment and mildly distored by recording imperfections, can be invaluable in the refinement of some processing techniques particularly autotracking. Recorded data both taken in flight or from a rig may also often provide the means of capture of fast or transient events for subsequent replay and analysis in a slower time frame.

The commissioning or integration process itself almost always progresses by a process of attrition. The common belief is that once the current problem or barrier to progress is removed everything will fall into place but, in the event, removal of the first problem reveals the next and so on, truly a situation analogous to a set of "Russian Dolls".

Persistence and tenacity are of course generally rewarded and the system credibility should rapidly improve. At this stage, if not before, some of the firmware and software will have been exercised, improved and refined on the rig. This is fine but the software people will have discovered the benefits of rig system access to test new ideas and directly resolve their problems on the radar itself. This factor has to be monitored closely otherwise excessive system rig time may be absorbed in esoteric software development tasks. Uncontrolled or over frequent changes to software, however well intentioned, may also undermine general system performance consistency. For example it has been known for a set of control settings to be chosen for a measurement programme, midway through which an apparently innocent change of software is introduced bringing about a subtle system reinterpretation of one of the control settings with consequent confusion and time wastage.

4. CHARACTERISATION

System characterisation activities begin to infuse into the workload at about this stage. Emphasis changes from the somewhat mechanistic and qualititative commissioning or integration tasks to the more quantitative nature of characterisation. The purpose here is to provide calibration and reference data as far as is sensibly attainable in the laboratory. The results obtained have to be compared with theoretical or modelled performance and should be retained for reference purposes in subsequent flight testing.

Comparisons with anticipated performance may well present the systems engineers with some shortfall to explain. Hopefully this is not serious but some reassessment of tolerances, error budgets and losses, both r.f. and processing, is all too frequently required before all the "missing dB's" can be largely accounted for.

At this stage the prototype system should approach a state of readiness for flight testing. There may be a little tidying up to be done and where sponsorship by a government agency is involved some form of acceptance test to be arranged and implemented. A stage report is usually advisable at this time.

5. FLIGHT TRIALS

Preparation and planning for flight trials will, by this point, have progressed to an advanced stage. Flight trialling is a very different situation to laboratory system testing. The latter, despite all the problems encountered on the way, offered a relatively clinical environment, time to think and, most importantly, opportunity to repeat an observation under virtually identical conditions. In the aircraft none of these luxuries can be relied upon, even in the most well appointed four engined transport. The equipment itself is less accessible and accurately controllable conditions are unattainable. Range and bearing geometry to a chosen contact, as a simple example, are continuously changing: no one can stop the aircraft to check a measurement.

In response to this situation one school of thought advocates extensive video recording and other data collection taken robot fashion in pre-planned rigid trials without scope for improvisation. Heavy reliance is then placed on post-flight analysis. This approach has merit particularly in any earlier study trials but in equipment design proving trials there is no substitute for direct observer deduction with a degree of on the spot improvisation to obtain conclusive evidence.

Where data collection and ground analysis methods are resorted to, control has to be exercised to both avoid overwhelming the analysts with excess data and to validate the relevance and quality of all recorded data before the termination of flying. An incomplete set of in-flight results found after grounding the equipment may prove a very expensive omission.

The foregoing case for system engineers participating as flight observers implies a further set of qualities in staff selection. This goes further than merely obtaining medical certification and possessing a capacity for quick thinking. This same fast thought processing will often have to take place in an adverse environment. Critical decisions and technical deduction may well be required several hours say into an all night trial carried out at low level in conditions which have buffeted the aircraft and its occupants without relief. Add to this the nausea-inducing odours which pervade aircraft from their hydraulic and lubrication systems, not to mention the indelicate aroma of hot soup and foodstuffs emanating from the galley (surely intended, on such occasions, for only the most masochistic of the full time professionals) and the value of a strong constitution is clear.

The observer's ordeal on a rough trip may not end on landing for, apart from possible post-flight checks and calibration, there will, if the trial is staged from a military base, be other commitments. First exposure of a new equipment to the relevant military is bound to provoke their intense curiosity. The trials crew are typically questioned at length, often, truth to tell, over drinks at a service bar! This liaison, whilst virtually specifying a social dimension to the job is not without worth to the project team for it provides valuable insight into the future operators' attitude and expectations. Later this type of experience can be put to good use in devising training schemes for the service user. It also helps to establish an alternative informal liaison channel through which later problems or minor improvements, particularly to software, may be discussed.

The first tangible hurdle at the flight trials stage is the actual installation. This may have been preceded by "mock up" checks, involving cable and waveguide runs together with weight and electrical dummy loading in which some preliminary flying could have taken place. In this, excellent support from full time field engineering is greatly appreciated. Ideally key members from the field team will have been present at the works during the characterisation stage to familiarise themselves with the new radar. This should materially improve the prospects of an elegant installation with minimal delay.

The value of good field engineering support, embodying knowledge and experience peculiar to their kind and not easily unearthed from textbooks, is high. The systems engineers may know the radar inside out but without guidance through the formalities and everyday jargon of the flying industry, in or out of the air, they would soon flounder in confusion.

Following installation some ground running is customary. Adequate ground supplies have to be arranged if use of main engines or an onboard generator is to be avoided. This ground running may be the first opportunity for radiation checks on a complete system. In such cases, with proper safety measures taken, a convenient object on or beyond the immediate horizon such as a spire or a water tower may be enlisted to confirm essential radar loop characteristics such as pulse compression or coherence.

Other checks verifying both the integrity of the equipment and the retention of characteristics following installation are performed in this ground phase.

Tests also have to be run to demonstrate that the radar is essentially hazard free both to the aircraft, its equipment and its occupants.

The first flights are also concerned with safety; followed soon, if all is well, with what are known as "shake down" activities. If the radar is some form of surveillance type with an approximation to real time display, much may be gleaned by informed observation. System areas requiring initial in-flight verification before pressing on with formal trialling include antenna stabilisation, host aircraft motion compensation, any serious BITE indications, display uniformity (i.e. background levels and false alarm rates) together with some qualitative indication of the detectability of valid targets of opportunity e.g. aircraft, ships or road vehicles. It should also be possible to gauge the approximate extent of clutter penetration and compare with expectation for the prevailing conditions. This period may reveal the need for limited modification and redesign but in general it should be seen as a learning phase.

All round confidence will evolve to the stage where formal trialling with cooperative targets commences. Again it is important initially to curb over ambitious aspirations and to keep situations simple. The disposition geometry should be straightforward, target manoeuvres limited and at each progression only a single aspect of either target or radar changed. At all times the situation should be constrained to the minimum number of variables. In this way the results obtained will be explicable and most probably pleasing and satisfactory. It is also at this stage that the real value of the laboratory characterisation is apparent. The earlier laboratory and ground results provide the foundation or reference against which airborne performance may be both judged and understood. A trial organised in this way from such a base will be less subject to crisis, under better control and better able to supply answers to criticism. A major hazard which is avoided by not embarking ill-prepared into complex radar trials is that not only are the problems less likely to be difficult to unravel but the resultant climate tends not to induce flawed theories and myths which can themselves become further obstacles to understanding and progress.

The tendency is for the truth of these remarks to intensify as the trial progresses from detectability aspects into the information extraction processes such as autotracking. Throughout a disciplined and documented record of the trials should be maintained. A minor problem to flight observers in this respect is the means of noting their own thoughts and jottings during flight. Handwritten notes in a standard engineers notebook which probably has to be hand held, are certainly unsuitable and an impediment to concentrated observation. Some form of shorthand, improvised perhaps, in a flip-over notepad is better while a pocket audio recorder is perhaps best. In a similar vein the systems observer requires an ability to retrieve general system information quickly if he is to react fast enough to an unexpected situation. A simple example could be the selection of an altitude adjustment to sight the surface horizon at an amended range. Backing up limited human memory could call for a library of specifications, computer print outs and other reference data to be carried on board. A simple pocket notebook can be packed with an amazing and evolving amount of general unclassified information to provide an excellent and compact aide-memoire.

As with the characterisation phase the trial's end and major intermediate stages should be marked with the issue of formal findings reports.

6. CONCLUSION

This paper has attempted to identify a general approach or philosophy to the commissioning and performance evaluation of a new airborne radar. The approach sets out to achieve an integral programme from building up the system through to the completion of flight trials. A methodical and informed process has been advocated which it is hoped gives an insight into the importance of progressing such a major activity from a firm and growing foundation. In this way good technical control can be maintained and the answers to criticism or unexpected system behaviour may be provided with authority.

It is essential to the approach that higher management understands and supports it. The systems engineer may well be subject to overtures to cut schedules or to think more positively in the hope that it will all work first time. The sense of urgency should be acknowledged but the basic message must be got through and retained. Otherwise the probability of debacle and associated escalating costs to the project and its reputation can be very high.

Microelectronics The Next Fifteen Years

Dean Wallace Physicist
Microelectronics Branch Code 3318
Naval Weapons Center
China Lake CA 93555
USA

This paper attempts to predict some future trends in microelectronics. The paper focuses on CMOS integrated circuit technology. CMOS is presently the leading integrated circuit technology, and all trends show that it will continue to dominate the integrated circuit market in the future.

I examine the processing of what I feel will be the two leading CMOS technologies of the future, twin tub,and silicon on insulator (SOI). These two technologies have the capability of giving maximum CMOS device performance. With device geometries shrinking the problems associated with scaling are introduced, and some possible solutions are examined. The paper is concluded with a brief description of some fundamental limits in integrated circuit technology.

This paper will attempt to provide a brief overview of semiconductor processing trends and semiconductor material trends. To start a look at where we have been is a good idea. Starting fifteen years ago bipolar transistors were widely being used and MOS devices were starting to be widely used in hand held calculators and products that required small power consumption. At the time a large device had 5000 transistors. Memory devices were just emerging as well as four bit microprocessors. In the 1960's the integrated circuit market was mostly comprised of bipolar transistors for both digital and analog applications. In the mid 1970's digital MOS overtook bipolar technology in the digital arena and with decreasing geometries MOSFET'S are also challenging in the linear arena. (Ref 1)

We have come a long way in the last fifteen years. We now have chips with hundreds of thousands of transistors; device geometries have shrunk below the one micron range. MOS devices operate in the 100 MHZ range when not to many years ago they only operated in the several hundred KHZ range. Bipolar technologies are approaching the GHZ range. GaAS is being used for very high speed applications i.e. a digital flip flop has had reported speeds of 18 GHZ. Technologies are being mixed, bipolar and CMOS on the same chip. People are looking at using GaAS on silicon to get the advantages of both technologies. Silicon on insulator (SOI) seems to be a promising technology. ASIC (application specific integrated circuits) technology has become common and device geometries in the one to three micron range have become quite common. Gate arrays with 50000 gates or more have become available on the commercial market. These events have led to a different design methodology as the systems become much more complex and the density of the chips increases. System designers and IC designers have become more dependent on computer tools to aide them in their work.

In this paper I am going to concentrate on CMOS technology. All trends point to the continued increase of the use of CMOS in the future. I will first give a quick overview of VLSI processing trends and a little insight to what the processing of the future might be. I will then talk about parasitics in CMOS devices since these are limiting factors in device performance. Then I will talk about two different CMOS processing technologies, (1) twin tub and (2) silicon on insulator and show how different processing techniques can help eliminate many of the parasitic problems. I will then examine first order scaling and show why first order scaling doesn't work in the sub micron range. I will finish by taking a brief look at some of the fundamental limits of device physics. These limits will include quantum mechanical limits and thermal limits.

Processing Trends

In this section I will give a brief overview of the following subjects,

1) epitaxy
2) diffusion
3) ion implantation
4) lithography
5) etching
6) multi level metal

This section is not necessary reading for the continuity of the paper but can serve more for a reference or as a simple review for people not familiar with microelectronics processing. A good starting point is epitaxy. Most epitaxy is done by chemical vapor deposition. As dimensions get smaller, molecular beam epitaxy will become more prevalent. The advantage of MBE is that it is a low temperature process which minimizes outdiffusion and autodoping. MBE allows precise control of doping, and more importantly complicated doping profiles can be done with MBE. Epitaxy can also be important for silicon on insulator MOS technology which will greatly increase speeds because of reduced parasitics. Because of the many problems associated with this a non epitaxial approach for providing single crystal silicon is being used. There are two common non epitaxial approaches, (1) deposit polysilicon on an amophous substrate, and recrystalize the silicon by a thermal means, (2) do a deep ion implant of oxygen which forms a buried layer of silicon dioxide and then recrystalize the silicon on the surface. Both of these methods have problems minimizing the defect density of the silicon.

The next subject that I want to discuss briefly is diffusion. Not much needs to be said about diffusion other than junction depths of 1000 angstroms can be realized but profile measurement techniques commonly used are only good for depths of about one micron. However the secondary ion mass spectroscope allows measurement of shallow junction profiles and is a powerful analysis tool.

The next topic is ion implantation and is very important to VLSI technology. Ion implantation is the bombardment of a target with energizied ionized atoms. This causes the atoms to be implanted below the surface. Typical energies are from 3 to 500 KeV resulting in the atom being implanted from about 100 angstroms to 10000 angstroms below the surface. A big advantage of ion implantation is the number of dopant atoms can be very well controlled and the depth profile can be well controlled. Ion implantation is used in all or practically all doping steps in present VLSI technology. In the future very shallow

junction depths will be seen in MOS technology, 100 angstroms or less. An interesting note is that since ion implantation is a kinetic process the eventual shallowest junctions are probably obtainable from thermal diffusion and not ion implantation (Ref 2). Although it will be a long time before ion implantation is not the practical choice.

The next section is on lithography. This is the defining process as far as future IC production goes. The prevalent lithographic process now in use is optical lithography. The reason for this is the high throughput of optical lithography. Good registration and the resolution is adequate for most present day commercial processes. Diffraction is the limiting factor for optical lithography and until recently one micron was believed to be the lower limit for optical lithography. But with the advent of tri level photoresists and contrast enhancement material 0.5 micron resolution has been achieved. Other ways to improve optical lithography is to go to smaller and smaller wavelengths. Deep UV and xray lithography are presently being examined. Xray lithography is promising because of the short wavelengths (around ten angstroms) but masking is a very diffucult problem that needs to be solved before xray lithography becomes practical. There is a real limit to optical lithography and non optical methods are becoming more common. On the non optical front electron beam lithography is very popular. In electron beam lithography an electron beam directly writes on the mask or the substrate. Therefore electron beam lithography gives better resolution than optical lithography. The limiting factor for electron beam lithography is the backscatter of electrons off the substrate, this is called the proximity effect. One step further is ion beam lithography which will reduce backscatter and thus give better resolution. Monte Carlo simulation shows ion beam lithography to have negligible backscatter and therefore shows no proximity effect.

The next subject is etching. I won't even discuss wet etching because it is not practical for VLSI technology where the dimensions are so small that an isotropic etch cannot be tolerated. Dry etching techniques are the predominant etching tools for the VLSI and ULSI era. Among these reactive ion etching is very good because it allows an anisotropic and a preferential etch. I also want to briefly touch on multilayer metal. As device geometries continue to shrink device interconnect becomes a real problem. A large amount of useable silicon can be consumed just by device interconnect. The way to get around this is multilevel metal. By going to two, three, four, and even five layers of interconnect the density can be greatly increased. But going to multilevel metal is not a trivial problem. Two layer metal is now common and three and four layer metal is being developed but there have been real yield problems associated with these. You need good interlevel dielectrics that are planarized, free from pinholes and have a low dielectric constant.

CMOS Parasitics

I will now discuss parasitic capacitances and resistances and latch up of CMOS devices. In order to maximize the performance of the devices these parasitics must be greatly reduced. An understanding of the parasitics associated with these devices will help us understand what will have to be done in the future to maximize device performance. Parasitic capacitances play a major factor in inhibiting the device speed of MOS circuits so it is paramount to minimize this capacitance. There are source substrate capacitance, drain substrate capacitance, gate substrate, gate drain and gate source capacitance. There are many factors that affect the capacitance but for our purpose all we need to know is that we must minimize it to maximize device performance. Also of great concern is interconnect capacitance and resistance thus leading to RC delays. As circuit density increases and line widths decrease and wiring resistance and associated delays become very critical. This leads us to multilayer metal schemes.

Two other effects that I want to mention are the body effect and latch up. In the region where the gate source voltage is greater than threshold voltage then the substrate bias is the voltage of the source minus the substrate voltage. As the substrate bias increases the associated channel substrate depletion layer increases. This traps more charge in this layer causing the channel charge to decrease. So this increases the threshold voltage which lessens current flow and thus makes circuits slower. This is called the body effect for obvious reasons. Latch up is caused when enough current is injected into the substrate to turn on two bipolar transistors an NPN and a PNP that are formed by the different dopings. Thus depending on substrate resistance in the parasitic path the current draw can be large and cause circuit failure. Both these effects can be minimized or eliminated with different processing techniques i.e. silicon on insulator.

CMOS Technology

As previously pointed out, CMOS technology seems to be the technology of the future. Therefore, in this section I am going to talk about the process integration of CMOS devices. I am not going to discuss single well technology since I am interested in optimizing both n channel and p channel devices. Therefore the two technologies that I will discuss are twin tub CMOS and SOI processes.

TWIN TUB

Twin tub allows the optimization of both p channel and n channel transistors. The general processing sequence for the twin tub process is to start with a n+ or p+ substrate. Put a lightly doped epitaxial layer on top of this to help prevent latch up. Form your n and p wells. Then a oxide is put down and patterened for the gates. A thin oxide is grown for the gates. Polysilicon is then deposited over the wafer and etched to form the gates. Then Boron and Phosphorous implantations form the n and p regions for the source and drain regions. The circuit is then passivated and contact cuts are made to the silicon and metal is deposited. The metal is etched and another dielectric is deposited, vias are etched and metal is again deposited. This multilayer metal could encompass two, three, or even four layers. As a final note, blocks of transistors are separated by deep trench isolation. By deep trench I mean trenches deeper than the well diffusions. The tubs within these regions abutt so there needs to be a Vdd and Vss contact to the n and p wells.

SILICON ON INSULATOR

SOI has the potential for being the fastest of the MOS devices. It is also the most diffucult to process and accordingly the most expensive. But changes in processing capability will eventually change

this. The processing goes as follows. Put a thin layer of silicon on an insulator. This can be sapphire or spinel or it can be an amorphous dielectric using new non epitaxial means. These non epitaxial means look very promising for the future. Then etch the silicon to form islands. The etch must be anisotropic since the separation of the islands will be much less than the thickness of the islands. Next the p islands are masked with photoresist and a phosphorous implant forms the n islands. The n islands are then masked with photoresist and a boron implant forms the p islands. Next a thin gate oxide is grown over the structures and polysilicon is deposited and etched to form the gates. Next the sources and drains are formed by selective masking of islands and phosphorous and boron implants respectively form the sources and drains of the p type and n type devices. Next a dielectric is deposited over the whole device and contacts are cut and metal is deposited. There will be multiple layers of metal in order to allow full utilization of the increased gate density. With the high density, planarization techniques become very important in making good step coverage possible.

Silicon on insulator is very attractive because of the following reasons;

1) High circuit density. One reason for the high circuit density is that there are no n and p wells as in the twin tub process.

2) The circuits are very fast because of the small capacitance. The small capacitance is due to the insulating substrate which leaves only a capacitive contribution from the walls of the source and drain. Also because of the insulating substrate leakage currents are negligible or non existent.

3) Latch up is eliminated.

4) The problems associated with body effect are eliminated.

SOI has the potential of being the leading CMOS technology of the future. The high speeds and the high densities make this technology very attractive. It also changes to 3D quite easily with devices sharing a common gate electrode.

First Order Scaling of CMOS Devices

If these devices are indeed the device of the future then the next question is how small can they be. I am going to give a rather quick overview of first order scaling and you can see how this can be extended. This first order scaling is based on the 'constant charge model' (Ref 3) that basically says that the operational characteristics of a device can be maintained if the device is scaled by a factor of A. The following is then scaled with this dimensional constant A.

1) vertical and lateral dimensions

2) voltages

3) doping concentrations

The actual effect is then,

device parameter	scale factor	
length	1/A	
width W	1/A	
gate oxide thickness t(ox)	1/A	
junctin depth x(j)	1/A	
substrate doping	1/A	
supply voltage V(dd)	A	
E field across oxide	1/A	
depletion layer thickness	1	
parasitic capacitance WL/t(ox)	1/A	
gate delay VC/I	1/A	
DC power dissipation	1/A*2	(*2=raised to the power of 2)
dynamic power dissipation	1/A*2	
power speed product	1/A*3	
gate area	1/A*2	
power density VI/A	1	
current density	A	
transconductance	1	(Ref 4)

Several things become immediately obvious from the above; first as we decrease channel length we will have to increase doping concentration in order to narrow the depletion region. You can see the current density scales linearily and the line widths of the conductors will be decreasing so electromigration becomes a real problem as does IR drops along these conductors. These problems can be pointed out with yet another table.

parameter	scale factor	
line resistance r	A	
line response rc	1	
normalized line response	A	
line voltage drop	1	
normalized line voltage drop	A	
current density	A	
normalized contact voltage drop	A*2	(Ref 5)

It becomes obvious that it will become harder to take advantage of the faster devices because of the

interconnect problems.

The first order scaling rules described above do not adequately explain device behavior in the submicron range. I will give a list of reasons why first order scaling is not accurate for submicron geometries and then explain how processing changes can help correct these probelms. At small geometries large doping concentrations are used to prevent threshold voltage falloff. These high concentrations decrease carrier mobility and increase the number of hot electrons (Ref 6). By hot electron, I mean an electron that has an energy that is more than a few kT above the Fermi level. When these hot electrons become injected into the gate oxide they cause gate oxide charging and consequently change the threshold voltage of the device. These hot carriers are the result of not scaling the power supply voltage and continuing to decrease the channel lengths of the device. A problem with scaling voltages to eliminate these effects in that as you scale down threshold voltages you are bringing the device "on" conductance closer to the device "off" conductance. This implies that power densities cannot remain constant but will have to be increased. Aslo as the devices become smaller and smaller we will be talking about depletion layers in the hundreds of angstroms. As electron mean free paths and depletion layers become about the same then the electrons can be accelerated through the thin layers without scattering thus obtaining very high velocities; these effects are called ballistic effects.

Also as doping levels are increased, there becomes a point when the gate oxide breaks before surface inversion can take place. This concentration is above 1E+19 cm*-3 (Ref 7). As a reference point the surface concentration for a channel length of 2000 angstroms is between 1E+17 and 1E+18 cm*-3. Another problem is interconnect related. As line widths and spacing are decreased the RC delay factor in increased. For large pitch metallization a parallel plate model for capacitance is fairly accurate, but as the metal runs are scaled (spacing is often scaled less than height and width) fringing effects become very improtant in adding to the total capacitance. The resistance is also increasing so we can show a large increase in RC related delays. This again points to the importance of multilayer metallization for integrated circuits.

Also of importance is carrier velocity saturation. As the channel length decreases the propagation delay scales linearily insted of as the square of the channel length. This velocity saturation occurs at 2E4 V/cm for electrons and 1E5 V/cm for holes (Ref 9). Therefore the carrier mobility for holes and electrons become nearly equal for short channel devices. This fact could eliminate the need for sizing of NMOS and PMOS transisors in the future.

The problems listed above are just some of the problems associated with the first order scaling model. Scaling is seldom uniformily done as might be suggested by the first order model. Often lateral dimensions are scaled more than vertical dimensions. This kind of scaling makes the device less prone to failures but also has adverse effects such as making the topography quite rough and hard to planarize. Many of the problems that are encountered in scaling will be solved by new materails and new IC processing techniques. Although the two processing technologies for CMOS mentioned above will be dominant. But there will be process variations to continually improve device performance. For instance a lot of work is being done to minimize hot electron effects. Lightly doped drains, doubly doped drains and doubly diffused drains that form a step. Even with improving technology there are still some fundamental limits of physics and this is what will be examined next.

Physical Limits

The physical limits for device size are set by quantum mechanics and thermodynamics. Quantum mechanics tells us that for each eigenstate of a system there is an assocaited energy and transition between states will have an associated radiation or absorption of energy. So we have a quantization of energy in physical systems (Ref 10). The lower size limit of a FET depends on the discreteness of charge and the wave nature of electrons. The wave equation is then given by U=e(ikx) and for the one dimensional barrier problem we will have three cases, E greater than V, E=V, E less than V. The case of transmission with energy less than V is a purely quantum mechanical effect and is called tunneling. For the conventional transistor to operate properly the current due to tunneling must be smaller than the other currents in the transistor.

Next from thermodynamics we can discuss the entropy of a system or from the second law of thermodynamics an increase of order in one part of a system is matched by an even greater increase of disorder in another part of the system. In other words entropy is alway increasing in the universe. So with this in hand we can talk about switching energies. To switch between an high and a low state the energy must be large as compared to thermal energy kT. Theoritically a minumum switching energy of kT is required. Whether a workable system can be made with this low energy is questionable. The above shows that the low voltage limit depends on the charge of an electron and thermal fluctuation.

Conclusion

This paper has concentrated on CMOS devices because I feel MOS technology will continue to be the technology of the 1990's and into the 21st century. I think I have shown that advances in processing technology and new materials will continue to shrink device dimensions thus increasing speed and density. This in no way implies that there won't be a demand for or improvements in other technologies. I don't think that there is any question that silicon will continue to be the most widely used technology into the next century. There will still be specific needs for the very high speed of gallium arsenide. There will be continued research into hot electron devices and quantum devices. But as CMOS speeds continue to increase the range of uses for this technology continue to expand.

References

1 S.M. Sze Ed., "VLSI Technology", McGraw-Hill, New York, 1983, p.3.

2 T.E. Seidel "Ion Implantation", in S.M. Sze Ed., "VLSI Technology", McGraw-Hill, New York, 1983, p.261.

3 R.H. Dennard et al., IEEE J. Solid State Circuits SC-9, 1974.

4 Neil Weste, Kamran Eshraghian, "Principles of CMOS VLSI Design a Systems Perspective", Addison-Wesley, Reading, 1985, p. 152.

5 Neil Weste, Kamran Eshraghian, "Principles of CMOS VLSI Design a Systems Perspective", Addison-Wesley, Reading, 1985, p. 152.

6 Dale M. Brown, Mario Ghezzo, Joseph M. Pimbley, "Trends in Advance Process Technology-Submicrometer CMOS Device Design and Process Requirements", IEEE Proceedings, Dec., 1986, p. 1679.

7 Neil Weste, Kamran Eshraghian, "Principles of CMOS VLSI Design a Systems Perspective", Addison-Wesley, Reading, 1985, p.155.
8 J. Matsunaga, N. Matsukawa, H. Nazawa, S. Kohyoma, "Selective Polysilicon Oxidation Technology for Defect Free Isolation", IEEE IEDM Tech Dig, Washington, DC, 1980, p. 565-568.
9 Dale M. Brown, Mario Ghezzo, Joseph M. Pimbley, "Trends in Advanced Process Technology-Submicrometer CMOS Device Design and Process Requirements", IEEE Proceedings, Dec., 1986, p. 1680.
10 Carver Mead, Lynn Conway, "Introduction to VLSI Systems", Addison-Wesley, REading, 1980, p. 346.

EXPERIENCE IN THE INTEGRATION OF HUMAN ENGINEERING EFFORT WITH AVIONICS SYSTEMS DEVELOPMENT

D. Beevis
Human Engineering Section
DCIEM,
1133 Sheppard Ave. West,
Downsview,
Ontario M3M 3B9
Canada

SUMMARY

Based on a review of human engineering activities in ten major acquisition projects, this paper outlines some conclusions aimed at facilitating the integration of human engineering activities with the development of advanced avionics. Conclusions are also drawn about the systems design and human engineering processes, and the role that mission, function, and task analyses can play in integrating human engineering and systems development activities. It is concluded that an approach which combines the interaction of hardware, software, and human functions is made especially necessary by the impact of advanced technology on the roles of human operators and maintainers, on the man-machine interface, and on the system development process itself. Finally it is argued that there is a need to establish standardized approaches to the application of human engineering in avionics system design.

INTRODUCTION

Human Engineering (HE) and Systems Engineering (SE) were introduced as formal disciplines in response to the technological advances of thirty to forty years ago. It was soon apparent that Human Engineering should be an integral part of Systems Engineering, as implied in one of the earliest SE texts (1). This lead to the concept of the "systems approach" to HE. The inclusion of HE in SE activities is reiterated in more recent publications (2,3).

As avionics improvements have introduced increasing levels of automation over the past forty years, so the human factor in systems performance has become increasingly important. Despite this, and despite the attention being paid to human engineering in some advanced projects, the integration of HE with the systems design process continues to be problematic. A review conducted at this Institute (4) concluded that it is more likely that the HE aspects of system design will be overlooked or neglected than incorporated. Discussing reasons for this lack of integration, a Panel at the 1984 NATO DRG Panel VIII Workshop on Applications of Systems Ergonomics to Weapon System Development recommended that case studies be compiled and studied for lessons learned (5).

This paper is a partial response to that suggestion. The application of HE in ten projects involving advanced avionics, or similar technology, in which DCIEM scientists were HE advisors to the procuring agency, is reviewed. The review examines the application of HE throughout the systems development process. The impact of advanced technology on the systems development process is discussed. Problems which arose at each stage are categorized as due to either management issues, or to HE procedures and techniques.

IMPACT OF ADVANCED TECHNOLOGY ON SYSTEMS DEVELOPMENT

The projects which were reviewed date from 1972 to the present. They include both development and evaluation/selection activities for 1 single seat, and 5 multi-place aircraft, and 4 similar combat information systems. All included advanced technology and complex man-machine interfaces.

A key feature of these interfaces is that the technology used has many more "degrees of freedom" than older technology. Therefore it provides more opportunities to make sub-optimal design trade-offs, especially in the interactions between different parameters. For example in electro-mechanical displays, which are reflective, contrast of the characters or legends is dictated by the selection of paint for the background and the markings, and the control of veiling glare on the instrument glass. Contrast on a similar CRT display is a function of characteristics such as brightness and intensity setting, dynamic contrast of the CRT, phosphor type, shadow mask characteristics, reflection from two or three intervening surfaces, and transmission characteristics of a notch filter.

Another feature of advanced technology is that it implies significant changes in the roles, functions, and tasks performed by the human operator. Such changes increase the importance of integrating HE activities from the outset of system development.

The man-machine interfaces associated with such advanced technology are extremely flexible. They can provide an extremely large amount of in formation to the operator, and they can change as a result of operator action or the operational or environmental situation. Therefore the operator must maintain not only a current mental model of the operational situation and of the system state, but also a model of the interface, and where he is in the multi-page representation of the system and environment. The design of these systems therefore requires a more thorough analysis than did more traditional interfaces. Unfortunately the availability of such interfaces, and their superficial similarity, encourages systems designers to quickly establish a design concept based on such equipment. This leads to the deferment, or neglect of important questions of concerning the roles, functions, and tasks of the operators, and exactly how the equipment will be used, what information will be displayed, when, and to whom, and what controls will be provided. The postponement, or neglect, of HE issues has repercussions for several aspects of system design, as will be discussed below.

HUMAN ENGINEERING AND SYSTEMS DEVELOPMENT

In the Canadian Forces the approach to the application of human engineering in acquisition projects is based on that used by the US services (6). In many cases the same work items and design standards are used. The current standard which specifies the approach to human engineering MIL-H-46855B (7) assumes a sequence of stages in system design development which parallels the general stages recommended for Systems Engineering (Table 1).

In our experience that recommended procedure is not always followed in practice. Seldom is there a systematic search for, and evaluation of, candidate system concepts, followed by system development. Rather, the preferred concept is often identified very quickly, based on precedent, and the remaining systems development effort is devoted to making that concept work. In this respect the process is much closer to what has been called the "ad hoc" or "direct" approach to systems design (8), rather than the "standard" approach which includes either theoretical or experimental modelling and simulation. Athans (8) has noted that the direct approach is often used for the design of the overall system, with the standard approach being used for sub-system optimization.

Although the direct approach is understandable in terms of the cost savings involved, it has obvious limitations, and those limitations are exacerbated by advanced technology. In particular it encourages the tendency to base human engineering decisions on solutions to previous problems, rather than analysis or experimentation. In reviewing the such problems in the context of the systems development process, the general project management headings of Technical Planning and Control, Systems Engineering Process, and Engineering Speciality Integration (3) were used.

Table 1. Recommended Stages in Systems Engineering and Human Engineering Analysis

Systems Engineering Hall - Ref 14.	Systems Engineering MIL-STD-499A	Human Engineering MIL-H-46855B
Problem Definition	Mission Requirements Analysis	Preparation of Scenarios and Mission Profiles
Value System Design	-	-
System Synthesis	Functional Analysis	Definition of Functions; Information Flow and Processing Analysis.
System Analysis	Allocation	Estimates of Potential Operator Capabilities; Allocation of Functions
System Analysis	Synthesis	Gross Analysis of Tasks; Analysis of Critical Tasks
Optimization	Optimization	Workload Analysis; Preliminary System and Sub-system Design; Equipment Detail Design; Studies, Experiments, Laboratory Tests; Procedures Development
Decision Making	Logistic Engineering; Life Cycle Cost Analysis	Test and Evaluation

TECHNICAL PROGRAM PLANNING AND CONTROL

This is defined as the "..management of those design, development, test, and evaluation tasks required" (3). In the projects reviewed, planning and control factors which influenced the integration of HE with SE in included management approach to HE, organization, staffing, planning, and scheduling.

Management Approach to Human Engineering

In 1961 Melton (9) noted that the concept of a systems approach to human engineering had less acceptance and was implemented less often than "...a casual examination of regulations, mission assignments, contract clauses, and research and development project statements might imply". He argued that this was in part because management had not fully adapted to the concept, and in part because not all human engineering specialists had the time or opportunity to experience the systems approach to HE. Our experience indicates that the same holds true today: many involved in project management have yet to adapt to the need for human engineering, and few engineers or systems designers have experience of successful applications of HE to several advanced projects.

The management approach to HE most frequently encountered is that it is a factor in the detailed aspects of interface and workspace design exemplified by the various design guides or "cookbooks", and covered under the headings of System, Sub-system, Equipment, Work Environment, and Crew Station Facilities Design in MIL-H-46855B (7). In half of the projects reviewed, the HE activities were associated with design efforts, rather than with SE efforts (Table 2). Such an approach does not ensure that the benefits of advanced technology will be realized, since, as indicated above, the human engineering analyses required to optimize system performance are required prior to the design stage.

Table 2. Human Engineering Involvement in Project Development

HE Effort	Project No. 1	2	3	4	5	6	7	8	9	10
Planning (HEPP)	√	-	√	-	-	-	√	√	√	√
Systems Analysis	√	-	√	-	√	-	√	?	-	√
Design	√	√	√	√	√	√	√	√	√	√
Test and Evaluation	√	-	√	-	√	-	√	√	√	T.B.D
HE Staff Qualified/Experienced	√	-	√	-	-	-	-	√	-	√

Organization and Staffing

If Human Engineering is to be applied as part of systems analysis, it seems clear that it should be part of the Systems Engineering function. Goode and Machol (1) addressed the organization of SE functions, and argued that although some organizations were not effective, several different types of organizations can work, depending on the personnel involved. The same appears to be true of the organization of the HE effort. When HE has been included as part of Systems Engineering, it has been one of the branches of the SE management tree. This locates the HE function in the right place, but it does not guarantee that HE issues are treated effectively because they are, in many instances, orthogonal to other engineering considerations. There is a need to integrate the human engineering contribution with other engineering efforts through the techniques and procedures by which design/development issues are handled.

Integration cannot be achieved merely by making systems engineers responsible for HE. Despite their similarity to systems engineering techniques, the current HE techniques seem to require training or experience. Yet in only four of the ten projects reviewed were the contractor's HE personnel trained or experienced. A variety of staffing arrangements were used, ranging from the senior mechanical draftsman being responsible for human engineering, to all engineers being responsible for human engineering. Neither of these extremes proved satisfactory because they did not ensure that those responsible for the HE function were knowledgeable in the latest developments, particularly in Function, Task, and Workload Analysis.

Three recent projects demonstrated the importance of that knowledge. Systems engineers given the responsibility for HE analyses such as Functional Analysis, Potential Operator Capabilities Report, Function Allocation, Task Analysis and Workload Prediction, had problems in understanding either the utility of the technique, or how it related to other engineering activities. In two projects they could not understand the utility of Functional Analysis, or of a review of Potential Operator Capabilities. In one of those projects they argued that since the operators perform all the tasks associated with current manual systems, any more advanced system was bound make those tasks easier. In the third project the HE activities were planned by one company of a consortium, for implementation by another company with no experience in HE. Those made responsible for HE argued that the Function and Task Analyses were implicit in the way the system had been conceived, and that no other analysis was necessary. In fact, no analysis of how the system would be operated or maintained had been performed; the contractor had no rationale for the operator and maintainer tasks, the interface design, equipment procedures or training plan. As a result changes identified as necessary through Test and Evaluation were implemented in a piecemeal fashion, with no understanding of their impact on system performance.

Convincing systems engineers of the value of such analyses, and obtaining analyses of an acceptable standard would have been easier if clear, comprehensive, worked examples of each technique had been available. Although the human factors literature contains examples of the more common techniques such as the Function Flow Block Diagram, no general purpose guide has been found which covers the development and use of all techniques.

Planning and Scheduling

Advanced technology has a significant impact on the planning and scheduling of human engineering activities. The tendency to design the man-machine interface without scheduling an adequate amount of effort in the analysis stage has already been mentioned. A more fundamental problem arises because it is possible to introduce advanced technology at a faster rate than it is possible to understand the human engineering issues involved, or how best to use that technology. For example recent work at this Institute shows that basic questions regarding the implementation of electro-optical colour displays are complex and poorly understood. Despite the lack of understanding, such displays are entering service in increasing numbers.

The overall pace of development of such technology must be anticipated. In two of the most recent projects reviewed, the need to have the emerging technology implemented in the "next" system, rather than missing the

opportunity by 10 to 15 years, resulted in technology implementation being rushed. There was a rapid transition from Concept Development to Design for Production, with insufficient exploration or development of the concept. Thus the operator's tasks were not analyzed thoroughly, and the impact of the new technology on operator roles, and the relationship between senior and junior personnel was not investigated. No man-in-the-loop simulation was undertaken, despite the obvious need for such experimentation.

The key to the successful scheduling of HE activities is the Human Engineering Program Plan (7, 10). The HEPP specifies the organization, scheduling, and extent of HE effort, and how that effort will be integrated with others in the systems development process. Somewhat surprisingly, four of the advanced technology projects reviewed did not use such a plan, and in one case the plan was scheduled for delivery 8 months into the project. The surprise is in part due to the fact that the HEPP is a standard contract requirement. The importance of the HEPP (or the attitude behind its use) is indicated by the use of the different HE techniques. Those projects which had an HEPP involved an average of 6 of the analysis activities; those without an HEPP involved an average of 2 HE analysis activities.

Continuity of Approach

The man-machine interface in advanced technology systems is characterized by a large amount of information, and a large number of control options. One example, the F-18 interface, has already been discussed in AGARD (11); over 200 menu options are available through 12 major display pages on two CRT displays, and the HUD display has 15 major modes. Such complexity makes it difficult to ensure the continuity of approach to the interface as a system is developed. Establishing and maintaining a rigorous application of rules for the use of spatial, colour, intensity, or symbolic display coding is extremely difficult, as others have noted (12).

In addition, display format "configuration control" is a major problem. In one recent case, carefully established conventions on symbology and display formatting were compromised by last minute changes to the software. In another case, what was to have been an off-line display became a primary display because of shortcomings in the capacity of the computer handling tactical information. Predictably the operators routinely complain about the human engineer who placed one of their principal displays in a scarcely accessible location. It is therefore important to maintain good records, not only of what rules were developed, but of the HE rationale for those rules.

SYSTEMS ENGINEERING PROCESS

Systems Engineering involves "the sequence of activities and decisions transforming an operational need into a description of system performance parameters and a preferred system configuration" (3). The HE process implied by MIL-H-46855B involves such a sequence of activities through analysis, design, test and evaluation. The overall approach used in such analyses has been outlined in the work of AGARD Avionics Panel Working Group 08 (13). It parallels that recommended for Systems Engineering (3, 14), as shown in Table 1, although it differs in the emphasis on analysis rather than mathematical modelling and optimization.

Table 3 shows the use made of the different analysis techniques for the ten projects reviewed. Only one or two techniques were used in some projects. Interestingly those projects involved the most technology development, for example the development of a digital data bus and general purpose man-machine interface. The majority of the projects used most of the standard HE techniques, but in different ways, and to different degrees.

Table 3. Human Engineering Analysis Activities in Individual Projects

HE Activity / Project No.	1	2	3	4	5	6	7	8	9	10
Mission Analysis	√	√	√	-	-	-	√	-	√	√
Definition of Functions	√	-	√	-	√	-	√	-	-	√
Information Flow & Processing Analysis	-	-	-	-	-	-	√	-	-	-
Estimates of Potential Operator Processing Capability	-	-	?	-	-	-	√	-	-	√
Allocation of Functions	√	-	√	-	-	-	√	-	-	√
Gross Analysis of Tasks	√	√	√	-	√	√	√	√	√	√
Analysis of Critical Tasks	√	-	√	-	-	-	√	-	√	-
Workload Analysis	√	√	√	-	-	-	√	-	-	√
Studies, Experiments, Lab Tests	√	-	√	√	√	-	-	-	√	-

Mission Analysis

Mission Analysis is equivalent to the SE stage of Problem Definition (14), or Mission Requirements Analysis (3). The lack of any Mission Analysis in 4 of the projects reviewed indicates that its utility is not fully understood. It is a vital prerequisite to HE analysis, because it sets the requirements for what the manned system is expected to do. Thus it should be done thoroughly, and from the perspective of a manned system. In one recent project, mission analyses were prepared to show the air vehicle transmission loadings. Pilot activity was implicit in those loadings, but the analyses could not be used as the basis for HE analyses without being reworked.

In conducting mission analyses it is essential to use realistic scenarios, and to describe the most demanding missions. Operational problems which have arisen in at least two of the systems reviewed are directly attributable to the use of unrealistic missions. For example, communication with other units is a significant factor in operator workload in many modern systems. In one project the original mission and task analyses had low levels of communication, commensurate with routine usage. Original estimates of workload were 3 on a scale of 1 to 5. Operational experience confirms those ratings for the missions which were analyzed, but missions which were not analyzed are being rated at "5 - excessive workload".

Definition of Functions

Function Definition is a logical extension of the process initiated with Mission Analysis, and again is directly compatible with SE activities. The HE activities associated with the Definition and Allocation of Functions are intended to ensure that the functions of the operators and maintainers are derived from systematic consideration of the mission requirements. This is not always done; five of the projects reviewed did not include a systematic Definition of Functions. In two of the projects where functions were defined the contractors had difficulty conducting an analysis of the functions currently performed by human operators. They also had difficulty conducting the analyses down to the required level of detail, where functions can be unambiguously allocated to man, hardware or software.

Analyzing the functions to be performed by the human operators is understandably difficult. Whereas Goode and Machol (1) viewed the primary objective of HE as optimizing the man-machine link, many human engineers hold the view that HE is the study of the human components of systems, and the integration of those components with other system components. These two viewpoints illustrate a fundamental problem in Human Engineering. Man is both a system user and a system component. The former viewpoint encourages an approach to system design which treats man as somehow external to the system - someone who receives information and makes inputs to the system. The latter viewpoint encourages an approach which integrates consideration of what the user does with what other system components do. Both attitudes are important, because human operators also perform functions such as supervision, checking, and training, by virtue of their being system components.

The development of the CP-140 Aurora aircraft, which is one of the projects reviewed, illustrates the importance of both attitudes. One of the most notable features of the Aurora is that the six tactical operators are seated together in a U shaped crew compartment. The advantages that were anticipated for that layout included facilitation of a team approach, task sharing, consultation, reversionary mode operation, crew rotation, crew interaction, on-the-job exposure to more senior tasks, crew proficiency training, and monitoring of crew performance. Of 418 potential tasks for two operators, 106 were judged to be facilitated by the adoption of the integrated compartment. Yet none of the functional analyses conducted for the Aurora, and none of the analyses for three other projects which have been reviewed, include functions which reflect such activities as consultation, training and monitoring. Those functions derive from their impact on the performance of the human operators, but do not derive from a Functional Analysis.

Allocation of Function

This stage of analysis is again directly compatible with the SE process. Geer (15) noted that there are three approaches to Allocation of Function: "trial and error" substitution of alternatives into a system or sub-system model; an evaluation matrix of plausible operator roles and equipment functions based on qualitative performance capabilities; an evaluation matrix using weighted performance scores for different functions. Only 4 of the projects reviewed used a formal approach to Allocation of Functions, and of those, only 3 are believed to have used such techniques (only two were documented). Other projects presumably used the traditional "ad hoc" approach to deciding the operator and maintainer functions. This is undesirable because there is a tendency for operators and maintainers to be allocated those functions which are not easily engineered. This accounts for the observation that most military roles involve either sensing, decision making, or complex, adaptive manual materials handling.

Notwithstanding current efforts (16), advanced technology makes it increasingly difficult to allocate functions on a rational basis. The capabilities of technologies such as Expert Systems or Direct Voice Input are difficult to estimate, and their optimum use to complement human capabilities and limitations is difficult to determine in advance of operational use. In our experience it is difficult for designers to envisage new functions, or for operators to envisage how new capabilities will be used and exploited. The trade-off of task complexity against selection and training is particularly difficult, despite implications to the contrary in the human factors literature.

One aid to such problems is the analysis of Potential Operator Processing Capabilities (7). That analysis reviews what operators might be able to do in a new system, in terms of their ability to process information. We have not experienced much success with it. In only 2 of the projects reviewed was there a systematic study of Potential Operator Processing Capabilities. In both cases they were contract requirements and in one of those cases the purpose of the analysis was misunderstood by the contractor. The contractor's report discussed the possible effects of environmental stress on operator performance, rather than the basic capabilities of the operators to perform anticipated functions. Yet it is the POPC Analysis which formally provides information on what operators and maintainers can be expected to do. It can make a significant contribution to two of the formal Function Allocation techniques (15), and assist in resolving problems about task complexity and training.

Task Analysis

Task Analysis describes the actions of operators (and maintainers), derived from the Allocation of Functions, for use in workload prediction, equipment design, equipment procedures development, training system design, and development of performance characteristics for Test and Evaluation. In fact the preparation of Task Analyses can serve as an integrating function for those developmental activities. It is undoubtedly a reflection of its utility that some form of Task Analysis was undertaken in 9 of the 10 projects reviewed. Unfortunately the analyses were not always carried out at the right time, or to the most effective level of detail.

In 5 out of 8 projects involving multi-function displays and controls, designers established the man-machine interface well before conducting detailed task analyses. In every case the final display requirements were underestimated. In three cases displays could not be optimized because of constraints imposed by those early decisions. In three cases the operational sequences required to use the displays and controls were unsatisfactory, being too complex or inefficient. In

one such case where the location of displays and controls had to be modified, a sub-contractor concluded after the event that the most effective approach to workplace design is to await the completion of system analysis and then design the workplace.

The operation of all of the interfaces of a system must be considered in such analyses. In one project one particular operator-machine interface was analyzed and used as the basis for the design of the whole multi-operator system. Subsequently some preferred display formats had to be modified to suit the constraints imposed by the general purpose interface. The premature decision on the interface also resulted in the adoption of a shadow-mask colour display for all functions, whereas the subsequent task analyses showed that high-resolution, monochrome displays were required for some of the system sensors.

Task Analyses must not only be timely, they must be complete. In one project Mission Analyses were not conducted, and Task Analyses were conducted for only the engagement sequences of the basic weapon. No analyses were conducted for the larger system of which the weapon is a part. As a result there is an on-going debate as to how best to handle information from other platforms, and whether operators will have the time to handle tactical information, or will be able to respond only to voice messages. The system was not designed as a true system, but as a number of independent units, with the assumption that the users will somehow make it work.

The majority of Task Analyses are conducted at a "gross" or "upper" level. Critical Task Analysis, as defined by (7), seems to be used rarely. In fact we have never seen a Critical Task Analysis which provided all the information required by MIL-H-46855B. When detailed task analyses have been produced they have usually been Operational Sequence Diagrams (OSDs) (13, 15). Five of the 10 projects used such an approach. OSDs do not provide all the necessary information, however, because they do not readily indicate the required performance standards, the impact of operator error, or the necessary job skills.

Analyzing the impact of operator error is increasingly important [illegible] roles of operators and maintainers change to those of a system monitor and supervisor. In one project, w[illegible] the development of a general purpose communication system, the contractor described the operation of the system using Signal Flow Graphs, or State Graphs. This technique is often used to describe communication systems. However the graphs were used to describe only the correct operational sequences. They did not describe incorrect sequences (the graphs become much more complex if this is done), and as a result they did not show that it was possible for an operator to dismantle a whole communications net if he made one particular error.

Workload Analysis

In only 4 of the projects reviewed was there a formal Workload Analysis. Again, the problems introduced by advanced technology require that far greater use is made of this technique. As others have noted (11, 17), advanced technology can add to the workload of the operator or maintainer. With its emphasis on information, advanced technology encourages systems developers to display "all possible" information to the user(s). This is typified by the development history of the HUD, Helmet Mounted Displays, and multi-function CRT displays. As others have pointed out (11, 17), such information is not usually integrated, and can increase operator workload unless the conditions under which it is used have been carefully defined.

In one recent project, the original concept had 34 pages of information; engineering developments increased that to 71, with a disproportionate increase in the complexity of the menu selection sequence. During trials it was observed that the senior operator, who had a more complex page selection menu, found it easier to slave his CRT display to the junior operator's, than to find his way about the display selection menu tree. The complexity of such systems adds to the operator's workload because he must not only maintain a current mental model of his operational environment and the systems he is controlling, he must also maintain a mental model of where he is in the multi-page representation of the system and environment.

Early attempts to use computers in Air Traffic Control resulted in increased workload as the controllers passed information to the computer by "induced tasks" (18). Similarly, multi-function controls and menus can increase the work required to input information by a series of selections. de Callies and Potter (17) report the case in which the change from dedicated controls to multi-function controls tripled the activity required to initiate a simple change in radio frequency. Such problems need not occur if given sufficient attention during system development. In one project a review of a complicated operational sequence led to a tenfold reduction in the number of individual actions required by the operator.

Advanced technology has an impact on the techniques used for Workload Analysis, because it changes the tasks performed by the operators and maintainers. The "classical" aerospace approach to workload prediction has been to calculate it from the ratio of time required to perform given tasks to time available. Such an approach works for behaviouristic, or mechanistic tasks such as selecting operational modes in response to information obtained by looking at a display. It is more difficult to apply in situations where the operator's tasks have a high cognitive content, or where the operator is task sharing. The "timeline" approach to workload prediction is therefore being replaced by other developments, several of which are based on the concept of attentional demand. To date, however, no one technique has widespread acceptance outside the organization which originated it.

Performance Specification

Few question the importance of expressing the functional requirement of a system in clear terms of effectiveness, but this seems to be done rarely. It does not appear to have been a clearly identifiable stage of analysis in any of the projects which were reviewed. In part this may be because it is difficult to show the impact of human performance, or the benefits of HE, prior to actually putting a system into operation. Advanced technology makes such predictions even more difficult, because it changes the standards of acceptable performance of systems, for example the sensitivity of sensors, or the response time of control systems.

Chapanis, in 1961, noted that the familiar measures of operator performance such as speed and error do not impress systems designers, particularly when compared with the cost and value estimates available from other specialities (19). The suggested remedies of conducting research using "systems relevant" criteria, as advocated by Meister, for example (20), or developing Operations Research (OR) techniques which incorporate human performance as a factor in system performance, have not been widely used.

We have had little experience of, or success with, OR type models. In only one of the projects reviewed was an OR type model of system performance used. The human performance characterized by the model made the usual assumptions that the human operator is completely reliable, is linear, stationary, and has a gain of 1. In one other project a contractor did attempt to use a model of human detection performance. Unfortunately the model was used to show that the human operator would improve the signal-to-noise ratio of his sensor system to a level where it met the specifications set for hardware performance, independent of the man-machine interface. Nevertheless recent developments in operator performance modelling appear promising. Network modelling tools such as SAINT promise to improve the ease of development of such models, and such an approach is being used to predict workload in the most recent project reviewed.

ENGINEERING SPECIALITY INTEGRATION

This topic covers "the timely and appropriate intermeshing of engineering efforts and disciplines such as ... human factors ... to ensure their influence on system design" (3). Several issues which modify the integration of human engineering with the system engineering process have been identified in the foregoing discussion. These include the approach taken to management and planning of the HE effort, and the lack of standardization in the use of available techniques. The main thrust of the remaining discussion is to identify promising solutions to some of those problems.

This review indicates that many of the factors which hinder the integration of human engineering activities with other systems development activities are, in large part, the old complaints of the effort being too little, too late. That this is still a problem is disappointing, because human performance factors have become much more critical as technology has advanced. As was predicted in 1959 (21), the task of improving the reliability of the human components of systems has become more important as the reliability of the machine components has improved; and as predicted in 1964 (22), the role of the human operator has changed, and research has not dealt with the complexity of real world roles and tasks.

Organization and Procedures

There are other factors which mediate the success of HE integration. In our experience, the most successful organizational means of integrating HE with SE activities was the Tactical Crew Compartment Review Committee that was formed to manage HE issues in the CP-140 Aurora project. The Committee's purpose was to integrate the activities and opinions of operators, maintainers, engineering specialities, and human engineers working on different aspects of the crew compartment. This appears to be the the function envisaged for the Human Interface Integration Team (HIIT), which was recently suggested as a means of moving HE functions away from a reactive "wrist slapping" role to a proactive design resource (23).

The CP-140 Committee was very successful as a forum for examining and reaching consensus on any issue which impacted the operation of the crew compartment. Much of the success of the committee was due to its role in fostering communication between different specialities: our experience in that respect supports the argument (2) that SE "can only be accomplished by an interdisciplinary team, and the first and most persistent problem of such a team is effective communication". Contrasting experience was provided by two projects where the HE interests were split up between management, operators, vehicle engineering, reliability and maintainability, and life support equipment specialists. The result was an ill-organized approach to human engineering which resulted in (sometimes heated) disagreements among the different interests.

A HE Co-ordination Committee is also able to facilitate designing for operational functionality. By this is meant that Functional Analysis is conducted from the viewpoint of how the system will be used, rather than from a concern of what it does. As this review has shown, the advanced man-machine interface is often dealt with "functionally" by providing for each role and task, eg. an active sensor display, a passive sensor display etc. or a display page for UHF radio control, another for VHF etc. But that approach does not ensure true functionality, because the way the operator will use the equipment over time has not been analyzed or refined. Indeed our experience supports the finding of Graham (24) that many designers do not know exactly how some controls and displays are used in practice. A functional approach which emphasizes how the system will be used couples the interaction of hardware, software and human functions, and leads to a more effective integration of human engineering with other engineering efforts. To do this requires an attitude that HE can and should contribute at the Function Analysis stage, and throughout the development process, including Task Analysis and Workload Analysis.

Techniques

If HE is to interact effectively with other engineering specialities early in systems development, more effort must be scheduled for HE analyses, tests and experiments to explore alternatives at the Allocation of Functions stage. This should include exploration of potential operator capabilities, and more detailed investigations of what operators can and cannot do with existing systems. In this context it seems unavoidable that more use must be made of man-in-the-loop simulation. Recent advanced aircraft projects in the USA and Europe have employed such simulation, but not all projects do so. Man-in-the-loop simulation is very expensive and time-consuming, and cannot be expected in all projects. Only one of the projects we reviewed made extensive use of it as a development tool. One possible solution to the problem of costs is to arrange for any Systems Integration Facility, or Laboratory to support man-in-the-loop experimentation.

An increase in the in the use of experimentation must be matched by an increase in the use of performance standards. It will also require improvements in performance measurement techniques within the context of systems operation. The author has personal experience of the benefits of using the system Parameters Document (25) as a means of specifying the human operator performance characteristics of a new system. It worked well, but the majority of entries proved to be design standards rather than performance standards, because no effort had been scheduled to derive operator performance standards for the system. The move to such "Parametric Documentation" which started in the 1960s does not seem to have been followed up, or fully exploited, although more recently the US Army has argued that there should be a shift from design specification to specification for performance in Human Engineering (26). The use of OR type models which incorporate human operator performance is seen as a promising approach to identifying the aspects of operator performance which are critical to system effectiveness.

A consistent theme of this paper has been that available HE techniques are not being used. This agrees with a previously reported conclusion that the basic problem is not one of lack of HE data or principles, but a lack of attention to their application (4). It is also apparent that there is a need to develop improved techniques. In 1967 Singleton, Easterby and Whitfield (27) argued that the increase in scale and complexity of systems required an equal advance in HE approach and techniques, and that the most glaring gap in current expertise was in the area of Task Analysis. In 1981 Topmiller (28) indicated that a range of techniques had been developed, many of them related to Operations Research, but that comparatively little use was being made of them. He argued for more effort to be devoted to technology transfer from the developers to the potential users. In 1984 the NATO Workshop on Systems Ergonomics concluded that many techniques were not "user friendly" and not easily transferred outside the laboratory where they were developed.

Standardization of available HE techniques applicable to systems development would facilitate their understanding and use by other engineering specialities. An initiative to do so was started within NATO MAS Aircraft Instruments Panel, in 1986, and such standardization is one of the aims of the recently formed NATO DRG Panel VIII Research Study Group on Human Engineering Analysis Techniques. The recommendation of the 1984 NATO DRG Workshop on Systems Ergonomics, that a NATO Clearinghouse be established to evaluate, standardize, certify, and make available human factors techniques, methodologies and findings, together with their documented applicability, generalizability and merit has yet to be acted upon.

CONCLUSION

One premise of this conference was that the typical approach to design may not be appropriate for the development of advanced, highly integrated avionics. This review has shown that advanced technology exacerbates many of the problems associated with the application of Human Engineering to systems development. It is concluded that these problems have their basis primarily in management issues, such as the attitude to HE, organization and staffing of the HE effort, and, perhaps most importantly, planning and scheduling that effort. It is also concluded that the approach that should be followed to improve the application of HE in advanced development projects is directly compatible with the recommended approach to Systems Engineering.

A variety of procedural solutions have been discussed. Perhaps the most important is that a Human Engineering Plan should be prepared at the outset of any developmental project, and that a coordinating committee with representation from operators, engineering specialities, and human engineering, is one of the most effective ways of achieving the integration of the different interests. It is also suggested that an approach to functionality which emphasizes how the system will be used on a temporal, or mission segment basis, can integrate not only the various Human Factors activities, but all engineering speciality efforts.

Finally it is concluded that some currently available tools and techniques can contribute to the successful application of HE. Chief of these is a properly conducted Task Analysis based on a realistic Mission Analysis. Two techniques which appear promising are the use of OR type models and man-in-the-loop simulation, to investigate the impact of operator performance on system performance. Neither technique appears to be widely used, however. It is suggested that man-in-the-loop simulation would be fostered if the Systems Integration Facilities, of Laboratories, which are being used increasingly in advanced projects, are designed to support HE tests and simulations. It is also concluded that there is a need for standardization of the techniques which are used to apply human engineering at all stages of project development.

REFERENCES

1. Goode, H.H., Machol, R.E., System Engineering, New York, McGraw-Hill, 1957.

2. Chambers, G.J., What Is a Systems Engineer? IEEE Transactions on Systems, Man, and Cybernetics, Vol SMC-15, No 4, July/August 1985, pp 517-521.

3. ________, Military Standard - Engineering Management, MIL-STD-499A (USAF), 1 May 1974.

4. Beevis, D., and Hill, Capt. M.W., The Designer as the Limiting Human Factor in Military Systems, NATO, Proceedings of the 24th DRG Seminar on The Human as a Limiting Element in Military Systems, May 1983, DS/A/DR(83) 170, pp 355-368.

5. Merriman, S.C., Muckler, F., Howells, H., Olive, B.R., Beevis, D., NATO, Applications of Systems Ergonomics to Weapons System Development - Volume II, September 1985, DS/A/DR(84)408.

6. Beevis, D., NATO, The Ergonomist's Role in the Weapon System Development Process in Canada, in: Applications of Systems Ergonomics to Weapon System Development - Volume I, April 1984, DS/A/DR(84)408.

7. ________, Military Specification - Human Engineering Requirements for Military Systems, Equipment and Facilities. MIL-H-46855B, 5 April 1984.

8. Athans, M, and Falb, P.L. Optimal Control, New York, McGraw-Hill, 1966.

9. Melton, A. in: Gagne, R.M., Psychological Principles in System Development, New York, Holt, Rinehart and Winston, 1962, Foreword.

10. Chaikin, G. The Department of Defense Human Factors Standardization Program, Applied Ergonomics, Vol 15.3, 1984, pp 197-201.

11. ________, Human Factors Considerations in High Performance Aircraft, AGARD Conference Proceedings, AGARD-CP-371, November 1984.

12. Merriman, S.C., and Moore, J.P., The F-18 - A New Era for Human Factors, in Human Factors Considerations in High Performance Aircraft, AGARD-CP-371, November 1984, pp 19-1 to 19-5.

13. Stringer. F.S., AGARD. Optimisation of Pilot Capability and Avionic System Design, 1978, AGARD-AR-118.

14. Hall, A.D., Three-Dimensional Morphology of Systems Engineering, IEEE Transactions on Systems Science and Cybernetics, Vol SSC-5, No 2, April 1969, pp 156-160.

15. Geer, C.W., Boeing Aerospace Co., Navy Manager's Guide for the Analysis Sections of MIL-H-46855, 1976, D180-19476-2, prepared for USNADC.

16. Hunt, G.R., et al., AGARD, Impact of Future Developments in Electronic Technology on Cockpit Engineering, Report of AMP to AVP Workshop on The Potential Impact of Future Developments in Electronic Technology on the Future Conduct of Air Warfare, in draft.

17. de Callies, R.N., and Potter, N.R., Development of Human Engineering Standards for Multifunction Keyboard Design, IEEE International Conference on Cybernetics and Society, Washington D.C., November 1976.

18. Innes, L.G., Man and Automation in Canadian Forces Terminal Air Traffic Control, DCIEM Report No 825, February 1972.

19. Chapanis, A. On Some Relations Between Human Engineering, Operations Research and Systems Engineering. in: Systems: Research and Design. D.P. Eckman, Ed. New York. Wiley. 1961.

20. Meister, D. Heresies: Brief Essays on Human Factors, Naval Personnel Research and Development Center, San Diego, Ca., March 1977.

21. Peters, G.A., Hussman, T.A., Human Factors in Systems Reliability. Human Factors, Vol 1, April 1959, pp 38-42.

22. Miller, R.B., Personal Communication to W.T. Singleton, quoted in: The Human Operator in Complex Systems. eds Singleton, W.T., Easterby, R.S., Whitfield, D., Taylor and Francis, London, 1967.

23. Shyler, J. Letter to "The Edge", Newsletter of the UCI (User Computer Interaction) Working Group. D.R. Eike ed. Carlow Associates Inc. September 1986.

24. Graham, D.K., JANAIR, Cockpit Switching Study - Phase 1 Final Report, JANAIR Report 730501, Boeing Aerospace Co, Seattle, May 1973.

25. __________, Preparation Instructions for Parameters Document, US NAVWEPS Ordnance Data 27070, 15 Jan 1963.

26. Promisel, D.M., Miles, J.L., and Cherry, W.P., Army Research Institute for the Behavioral and Social Sciences, Human Factors, Manpower, Personnel, and Training Required Operational Capability (ROC) Enhancement, ARI-RP-84-23, 1984.

27. Singleton, W.T., Easterby, R.S., Whitfield, D., The Human Operator in Complex Systems, Taylor and Francis, London, 1967.

28. Topmiller, D.A., Methods: Past Approaches, Current Trends and Future Requirements, in: Morall, J., Kraiss, K-F, Manned System Design: Methods, Equipment, and Applications, NATO Conference Series, New York, Plenum Press, 1981.

LE TEST DE LOGICIELS AVIONIQUES COMPLEXES :

UNE EXPERIENCE PRATIQUE

M. MUENIER

ELECTRONIQUE SERGE DASSAULT

92214 SAINT-CLOUD - FRANCE

RESUME : Cet exposé présente les techniques utilisées à l'Electronique Serge DASSAULT pour le test de logiciels avioniques, principalement des logiciels embarqués utilisés à bord des avions MIRAGE F1 et MIRAGE 2000.

Les exigences de sûreté de fonctionnement qui s'attachent à de tels logiciels imposent des contrôles rigoureux à tous les stades de développement et particulièrement à celui du test. Elles ont aussi amené à concevoir et à utiliser des méthodes et des outils, dans une approche globale du cycle de vie du logiciel.

C'est pourquoi l'accent sera mis sur l'impact que peut avoir sur le test l'actuelle évolution des méthodes de spécifications, et l'intégration croissante des outils.

INTRODUCTION

Les applications avioniques, et de façon plus générale les applications embarquées, se caractérisent par un certain nombre de propriétés telles que leur longue durée de vie et des contraintes sévères en volume et en poids.

Un certain nombre de facteurs, propres à ces applications, ont un impact particulièrement important sur les activités de test. C'est, en premier lieu, le haut niveau de sûreté de fonctionnement requis qui, à lui seul, justifie la part importante prise par les tests dans le processus de développement (de 40 à 50%). C'est, ensuite, l'aspect "embarqué" qui, de par la spécificité des calculateurs cibles et de leurs entrées-sorties, conduit à des développements "croisés" et impose, pour les tests menés sur calculateurs cibles, de simuler l'environnement opérationnel. C'est, dans le même ordre d'idées, l'aspect "temps-réel" qui exprime l'obligation dans laquelle le logiciel se trouve de réagir rapidement à des événements extérieurs. Cette caractéristique "temps-réel" impose, au niveau du test, une reproduction aussi fidèle que possible des aspects temporels tant de l'environnement que du comportement du logiciel lui-même (exécution du logiciel réel non instrumenté).

D'autres facteurs, moins spécifiques, ont également un impact important sur le test de logiciels avioniques ; ce sont, par exemple, le taux élevé de modifications (tests de non régression) et le souci de productivité (tests symboliques).

Tous ces facteurs font du test des logiciels avioniques une opération difficile et coûteuse, pour laquelle peu d'outils étaient encore disponibles il y a quelques années. On a pu dire ainsi que les applications temps-réel constituaient le 'Monde Perdu' du test et de la mise au point de logiciel (Robert L. GLASS, Boeing Aerospace Company).

Nous présentons ici les solutions qui, à l'Electronique Serge DASSAULT (ESD), ont été apportées à ces problèmes depuis plusieurs années.

CHAPITRE 1 : L'EXPERIENCE DE L'ESD

L'ESD est spécialisée dans l'étude, le développement et la fabrication de matériels électroniques dans les domaines civils et militaires.

L'effectif de l'ESD est actuellement de 4000 personnes, dont plus de 2000 ingénieurs et cadres. Le chiffre d'affaires 1985 s'élevait à plus de 3000 MF (un tiers environ étant réalisé à l'exportation et un quart dans le domaine de l'informatique aérospatiale).

L'ESD est fournisseur de nombreux équipements des avions MIRAGE F1 et MIRAGE 2000 d'Avions Marcel DASSAULT-Breguet Aviation (AMD-BA) pour lesquels elle a développé plusieurs gammes de calculateurs embarqués universels (calculateurs M182 pour le MIRAGE F1 et série 84 pour les MIRAGE 2000). ESD fournit également le système de transmission d'informations numériques DIGIBUS qui est normalisé en FRANCE pour les trois Armes (norme GAM-T-101). L'ESD développe également depuis 1977 les logiciels opérationnels fonctionnant sur ces calculateurs ainsi que les logiciels de production et de test associés. Elle s'est dotée, pour répondre à ces besoins, d'importantes équipes logiciel (plus de 200 personnes).

Plus de 20 millions d'octets de logiciels opérationnels ont été ainsi livrés sur un ensemble de quinze projets. Pour chaque projet, une livraison est effectuée environ tous les mois ; elle comporte en moyenne 600 000 octets de code, 60 000 pages de listings et 10 000 pages de documents.

Ces logiciels sont à 90% écrits en LTR, un langage haut niveau temps-réel dérivé d'ALGOL. LTR fut l'un des premiers langages de ce type à être utilisé pour des applications avioniques puisqu'il a été mis en service à l'Electronique Serge DASSAULT en 1977. Les 10% de logiciel restant correspondent à des sections critiques en temps de calcul et sont écrits en langage d'assemblage. Très bientôt l'ESD disposera également, pour ses calculateurs, d'outils de programmation en Ada.

Il faut souligner l'effort important entrepris par l'ESD depuis 10 ans dans le domaine du génie logiciel qui s'est traduit, d'une part par la définition et la mise en application de la méthodologie MINERVE (1), d'autre part par la réalisation d'outils, principalement d'aide à la spécification (DLAO (2)) et au test (BVL (3), IDAS (4)). Le paragraphe suivant présente succinctement la méthodologie MINERVE.

(1) MINERVE : Méthodologie INdustrielle pour l'Etude, la Réalisation et la Validation de logiciel d'Equipement
est une marque déposée de l'Electronique Serge DASSAULT.

(2) DLAO : Définition de Logiciel Assistée par Ordinateur
est une marque déposée de l'Electronique Serge DASSAULT.

(3) BVL : Baie de Validation de Logiciel.

(4) IDAS : Information de la Détection d'Anomalies dans les Systèmes
est une marque déposée de l'Electronique Serge DASSAULT.

CHAPITRE 2 : LA METHODOLOGIE : MINERVE

La méthodologie MINERVE a déjà fait l'objet d'une présentation dans le cadre d'un symposium AGARD (J. PERIN : LOGICIEL AVIONIQUE : EXPERIENCES PRATIQUES D'UNE METHODOLOGIE - AGARD Conference Proceedings n° 272 - 1979). Nous rappelons ici ses éléments principaux.

MINERVE a pour objectifs de faciliter la production d'un logiciel de qualité dans des conditions de coûts et de délais maîtrisables.

La qualité d'un logiciel se définit non seulement par sa conformité aux spécifications mais aussi par sa facilité d'évolution, sa clarté, la précision de sa documentation, sa sûreté de fonctionnement, ses performances, etc.

La maîtrise des coûts et des délais réside dans la possibilité d'entreprendre à tout moment des actions tendant au respect des engagements. Elle s'obtient par la connaissance permanente de l'avancement du projet et des travaux restant à effectuer.

Pour atteindre les objectifs précédents, MINERVE, pièce maîtresse de l'assurance et du contrôle-qualité logiciel au sein de l'ESD, s'appuie sur trois principes :

- les projets sont découpés en phases et en étapes caractérisées par des activités, des produits et des responsabilités clairement définis.

- la qualité des produits ainsi que leurs coûts et délais de réalisation sont contrôlés de façon continue.

- les modifications sont prises en compte quel que soit le degré d'avancement des travaux, selon une procédure unique destinée à éviter toute dégradation de la qualité du logiciel.

2.1. Principe du découpage des projets

Les projets sont découpés chronologiquement à deux niveaux (les phases et les étapes) selon le schéma suivant :

PHASE 1. DEFINITION

Etape 1.1 : Définition globale du système
Etape 1.2 : Définition opérationnelle du logiciel
Etape 1.3 : Définition fonctionnelle du logiciel

PHASE 2. PLANIFICATION

Etape 2.1 : Contrôle de la faisabilité technique
Etape 2.2 : Définition des moyens techniques,
Etape 2.3 : Réexamen du plan-qualité logiciel
Etape 2.4 : Réexamen des plannings et des coûts

PHASE 3. REALISATION

Etape 3.1 : Conception globale
Etape 3.2 : Conception détaillée
Etape 3.3 : Codage et tests unitaires
Etape 3.4 : Tests d'intégration
Etape 3.5 : Tests fonctionnels
Etape 3.6 : Validation du logiciel

PHASE 4. EXPLOITATION

Etape 4.1 : Intégration du système
Etape 4.2 : Suivi du logiciel

Les étapes 1.1, 1.2 et la phase 4 relèvent de la responsabilité du maître d'oeuvre du système d'armes (AMD-BA), toutes les autres étapes relèvent de celle du réalisateur du logiciel (ESD).

Les ETAPES sont caractérisées par la nature de l'ACTIVITE PRINCIPALE qui y est exercée. Celle-ci se concrétise toujours par un ou plusieurs produits formalisés ou PRODUITS MINERVE : documents, cassettes, bandes magnétiques, etc. Une activité principale terminée, ses produits ne peuvent être modifiés que par une procédure particulière (cf. paragraphe 2.3).

2.2. Principe des contrôles

Les contrôles s'exercent, tout au long du projet, à deux niveaux :

- qualité des produits (programmes et documents),
- coûts et délais.

2.2.1. Contrôle de la qualité

Tout contrôle de qualité s'effectue sur un produit MINERVE.

Ces contrôles sont de trois types :

- Contrôles de type A :

  Ce sont des contrôles internes au produit d'une étape. Ils consistent à vérifier que le produit de l'étape respecte les règles spécifiques (précisées dans le plan-qualité logiciel) et les standards généraux (définis dans le manuel-qualité logiciel) qui lui sont applicables.

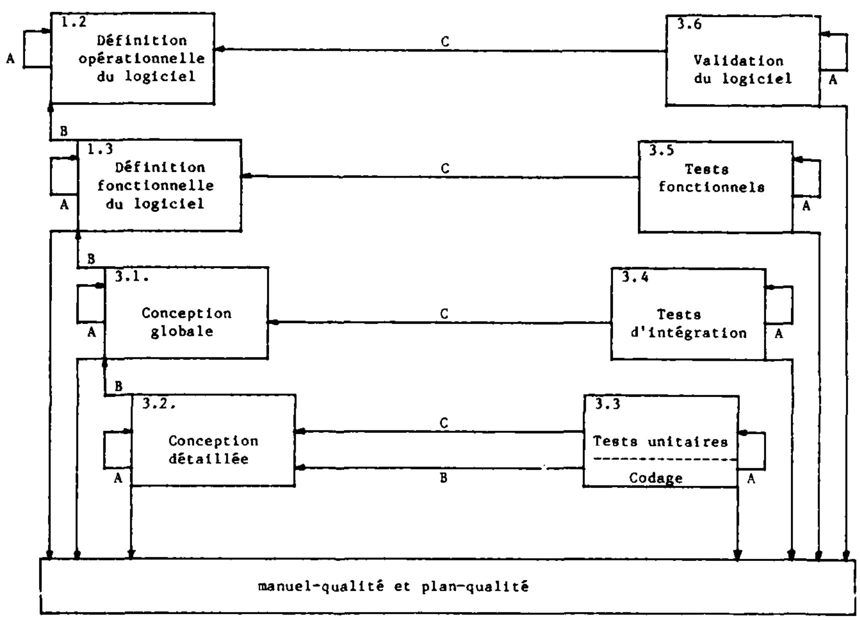
- Contrôles de type B :

  Ce sont des contrôles de cohérence entre les produits d'une étape et ceux des étapes antérieures. Ce type de contrôle, comme le précédent, est réalisé sous forme de relectures de documents et de revues de projet.

- Contrôles de type C :

  Ce sont les tests des programmes. Ils se déroulent en quatre étapes successives et s'effectuent pour chacune d'entre elles selon des points de vue et par référence à des documents différents. Ainsi les tests réalisés au cours des étapes de codage et tests unitaires (3.3), tests d'intégration (3.4), tests fonctionnels (3.5) et validation du logiciel (3.6) permettent de vérifier la conformité des programmes à leurs descriptions successives établies au cours des étapes symétriques : conception détaillée (3.2), conception globale (3.1), définition fonctionnelle (1.3) et définition opérationnelle (1.2).

La figure 1 schématise ces contrôles.



2.2.2. Contrôle des coûts et des délais

Ce contrôle s'effectue à intervalles rapprochés pendant toute la durée du projet. Il consiste à mesurer le degré d'avancement des travaux en coûts et délais, puis à le comparer aux prévisions consignées dans un document de référence.

2.3. Principe des modifications

La prise en compte d'une modification peut intervenir quel que soit le degré d'avancement du projet.

Aucun produit MINERVE ne peut être modifié en dehors de la procédure décrite ci-dessous, indispensable pour conserver la cohérence des différents produits pendant toute la vie du logiciel.

- Prise en compte de la modification :

  Après détermination du produit MINERVE à modifier situé le plus en amont dans le processus de développement du logiciel, une FICHE DE MODIFICATION de ce produit est rédigée par le demandeur si elle concerne les produits de l'étape 1.2, ou par le réalisateur si elle concerne les produits d'une étape ultérieure.

- Réalisation de la modification :

  La modification ainsi prise en compte conduit à la reprise de toutes les étapes dont les produits sont remis en cause, en commençant par la plus en amont de celles-ci.

L'exécution des modifications induites dans les différents produits concernés est consignée sur une FICHE SUIVEUSE, élaborée dès la décision de réalisation.

CHAPITRE 3 : LES TESTS DE LOGICIEL

Comme il a été vu dans le chapitre précédent, le test du logiciel intervient dans les étapes 3.3 à 3.6 du cycle de vie. A l'issue de ces étapes, le logiciel, validé sur son matériel, est remis au maître d'oeuvre du système d'armes qui procède alors à l'intégration du système (étape 4.1). Ce chapitre ne traite que des tests de logiciel proprement dits.

Lors des étapes 1.2 à 3.2, le système a été décomposé à différents niveaux (fonctions opérationnelles, fonctions logicielles, modules, pièces, etc..).

A chaque étape, les tests portent donc sur le niveau de décomposition décrit dans l'étape "symétrique" (cf. figure 1) selon un processus de recomposition progressive du logiciel en "blocs" de taille croissante. Cette approche présente l'avantage de faciliter à tous les stades la simulation de l'environnement.

Au fur et à mesure que des ensembles plus importants de logiciel sont testés, l'explosion combinatoire conduit à abandonner le test d'informations internes aux programmes (telles que les chemins de contrôle) au profit de tests fonctionnels qui deviennent plus significatifs. Les techniques de test correspondantes sont habituellement désignées par les termes de test "boîte blanche" et test "boîte noire".

Ceci est résumé par la figure 2 :

ETAPE		TEST BOITE BLANCHE	TEST BOITE NOIRE
3.3.	Tests Unitaires	Tests de branches Tests de chemins	Tests externes de pièces
3.4.	Tests statiques d'intégration	Tests de branches Tests d'interfaces entre pièces	Tests externes de modules
	Tests dynamiques d'intégration	Tests d'interfaces et de synchronisation entre processus	
3.5.	Tests fonctionnels		Tests externes de fonctions logicielles
3.6.	Validation		Tests externes de fonctions opérationnelles

figure 2

Il faut souligner que l'activité de test est associée à une activité de mise au point. Cette dernière vise à localiser l'origine des erreurs détectées par le test et nécessite, quel que soit le niveau, l'exploitation d'informations internes au programme (approche de type "boîte blanche").

Les techniques et les outils de test utilisés pour chaque étape (de la responsabilité du réalisateur de logiciel) sont présentés dans les paragraphes suivants.

3.1. Les tests unitaires

Le niveau de décomposition concerné est la "pièce" dont les caractéristiques sont d'être d'un volume réduit (inférieur à 100 instructions), de correspondre à une structure du langage de programmation possédant un seul point d'entrée et un seul point de sortie (par exemple procédure) et d'être une unité de compilation et d'archivage.

L'objectif principal des tests unitaires est de vérifier que la logique de contrôle des pièces est conforme à la définition qui en a été faite lors de l'étape de conception détaillée, et qui est consignée dans le DOSSIER DE CONCEPTION DETAILLEE DU LOGICIEL, produit MINERVE de cette étape.

Trois types de tests doivent être effectués sur les pièces (cf. figure 2) :

- Tests de branche :

  Les jeux de tests doivent être élaborés afin de provoquer l'exécution au moins une fois de chacune des branches de la pièce, en s'assurant après exécution que l'état de la pièce à la sortie de la branche est bien celui attendu.

- Tests de chemin :

  Un chemin est un ensemble de branches reliant logiquement le point d'entrée d'une pièce à son point de sortie. Les jeux de tests doivent permettre de parcourir chacun des chemins de la pièce en s'assurant à l'issue du parcours que la pièce est bien dans l'état attendu.

- Tests externes de pièce :

  Il s'agit de s'assurer que la pièce accomplit bien la fonction (ou les fonctions) prévue(s). Le jeu de tests doit comporter en entrée des données cohérentes et ayant une signification fonctionnelle.

La mise en oeuvre de ces tests est décrite, lors de la phase de conception détaillée, sous forme de fiches de tests unitaires.

Ces fiches contiennent trois types d'informations :

- les données d'entrée,
- les consignes de mise en oeuvre (lancement et trace),
- l'état attendu après exécution.

Lors de l'étape de tests unitaires, ces fiches sont traduites sous forme de programme de tests exécutés par l'outil IDAS (cf. Chapitre 4), un programme étant associé à chaque pièce.

Pour effectuer ces différents types de tests, il peut être nécessaire de simuler l'environnement de la pièce, soit par des pièces déjà testées, soit par des pièces spécialement écrites à cet effet.

Les volumes mémoire et les temps d'exécution sont également conservés afin de maîtriser ces paramètres au cours du processus d'intégration.

A l'issue de ces opérations, les produits MINERVE élaborés sont :

- les PIECES DE CODE sauvegardées sous forme de code source et de code objet,

- les LISTINGS de ces pièces,

- le DOSSIER DE CERTIFICATION DES PIECES constitué des fiches de tests unitaires renseignées lors de l'exécution de ceux-ci.

3.2. *Les tests d'intégration*

Le niveau de décomposition concerné peut être le processus ou le module.

Un processus correspond à l'ensemble des traitements dont l'exécution est subordonnée à la même condition d'activation (événement externe ou interne, fréquence).

Un module est une partie de processus possédant une cohérence fonctionnelle, c'est-à-dire appartenant à une même fonction logicielle (cf. paragraphe 3.3.). Un module est constitué d'une partie interface qui contient les informations visibles de l'extérieur ainsi que les directives d'importation de données de l'extérieur et d'un corps comportant des données locales et une séquence d'appels de pièces. Les langages de programmation les plus récents (LTR3, Ada) permettent maintenant de représenter très simplement les structures de processus et de module.

L'objectif principal des tests d'intégration est la vérification des interfaces entre les constituants (pièces, modules, processus) du logiciel, vis à vis du DOCUMENT DE CONCEPTION GLOBALE DU LOGICIEL (produit MINERVE de l'étape de conception globale).

Les tests qui sont conduits à cet effet sont de deux types :

- Les tests d'INTEGRATION STATIQUE : les interfaces entre pièces et entre modules, à l'intérieur de chaque processus, sont vérifiées indépendamment des contraintes 'temps réel'.

- Les tests d'INTEGRATION DYNAMIQUE : les différentes conditions d'activation sont mises en oeuvre afin de vérifier les interfaces entre les processus et l'environnement extérieur d'une part et entre les processus eux-mêmes d'autre part. Ces tests s'exercent, en particulier, sur le partage du temps (durée des cycles, synchronisation,...) et des données (cohérence, accès aux variables partagées,...).

Un premier travail consiste en la rédaction du DOSSIER DES TESTS D'INTEGRATION DU LOGICIEL. Celui-ci se présente sous la forme de FICHES DE TESTS STATIQUES de modules et de processus ainsi que de FICHES DE TESTS DYNAMIQUES de processus.

La formalisation de ces tests s'effectue ensuite de façon analogue à celle des tests unitaires par écriture et archivage de programmes de test à l'aide de l'outil IDAS. Deux outils supplémentaires sont utilisés.

. COCODIN (1), pour le contrôle du passage des paramètres, car la version du langage LTR utilisée actuellement ne dispose pas de la compilation séparée (cet outil ne sera plus nécessaire pour les programmes écrits dans la dernière version du langage (LTR3) ou en Ada). Cet outil recherche l'ensemble des procédures appelées par un module et les insère dans le flot de compilation de celui-ci. Le contrôle est alors effectué par le compilateur LTR.

. EDL (2) pour le contrôle des entrées-sorties de l'équipement lorsque le module ou le processus testé émet des informations sur celles-ci. EDL permet l'observation et l'enregistrement des messages émis sur un bus multiplexé de type DIGIBUS et le contrôle a posteriori des contraintes de temps et d'ordonnancement imposées à ces messages.

Les produits MINERVE de cette étape sont :

- le DOSSIER DES TESTS D'INTEGRATION DU LOGICIEL,

- un PROGRAMME sauvegardé sous forme de code objet et conforme au document de conception globale,

- le DOSSIER DE CERTIFICATION DES MODULES ET DES PROCESSUS constitué des fiches de tests statiques et dynamiques renseignées lors de l'exécution de ceux-ci.

3.3. Les tests fonctionnels

L'objectif de cette étape est de vérifier la conformité du programme, produit de l'étape précédente, au document de SPECIFICATIONS FONCTIONNELLES DU LOGICIEL (produit MINERVE de l'étape de définition fonctionnelle du logiciel).

Ce travail s'effectue en deux temps :

- tout d'abord, le DOSSIER DES TESTS DES FONCTIONS LOGICIELLES est rédigé. Ce dossier comprend, pour chaque livraison de programme, un ensemble de tests clairement identifié. Ces tests sont regroupés par "fonctions logicielles".

Les fonctions logicielles sont les fonctions du calculateur vues sous l'angle du réalisateur du logiciel de l'équipement.

Elles sont constituées par des regroupements de traitements selon des critères fonctionnels propres au réalisateur. Les critères de regroupement peuvent être :

. la facilité d'appréhension du logiciel,

. la facilité d'évolution,

. la facilité d'organisation du logiciel,

. la réutilisabilité.

Chaque test comprend :

. la séquence des actions nécessaires à son exécution, décrites d'un double point de vue fonctionnel et de mise en oeuvre. Le point de vue fonctionnel permet d'effectuer la corrélation des tests avec les spécifications fonctionnelles. La description de la mise en oeuvre est basée sur les outils utilisés : principalement la BVL (voir Chapitre 4) qui réalise la simulation de l'environnement de l'équipement et permet d'observer son comportement au travers des liaisons opérationnelles (test boite noire) et IDAS (voir Chapitre 4) qui permet d'effectuer la mise au point en observant le comportement temps réel et les informations internes du programme testé.

. Pour chaque action, les contrôles à effectuer décrits selon un double point de vue identique au précédent. Pour chaque système d'armes, ce contrôle utilise un ensemble d'"images" définies au niveau de la BVL. Ces images peuvent simuler un équipement (par exemple visualisations "tête haute" ou "tête basse") ou simplement regrouper des informations possèdant une cohérence fonctionnelle.

- Ces tests sont ensuite exécutés et leur résultat est consigné, constituant ainsi le DOSSIER DE CERTIFICATION DES FONCTIONS LOGICIELLES.

Pour chaque fonction logicielle, trois types de tests sont effectués :

- des tests nominaux, correspondant aux cas de bon fonctionnement du système d'armes,

- des tests de pannes, correspondant aux cas prévus lors de la définition,

- des tests de pannes aléatoires, visant à évaluer la robustesse non seulement du programme, mais aussi des spécifications fonctionnelles et pouvant ainsi conduire à remettre en cause ces dernières.

(1) COCODIN : COntrôleur de COhérence D'INterface
(2) EDL : Espion De Laboratoire

Lors de l'exécution de ces tests, l'environnement de chaque fonction est simulé à un double niveau :

- au niveau des interfaces équipement, en utilisant la BVL

- au niveau des interfaces avec les autres fonctions logicielles, en utilisant les fonctions logicielles elles-mêmes, préalablement testées et certifiées. Ceci impose de conduire les tests fonctionnels dans un ordre précis, prenant en compte les dépendances des fonctions logicielles entre elles.

Cette simulation ne porte que sur les interfaces nécessaires à la mise en oeuvre des tests ; en particulier la simulation des autres équipements du système d'armes n'est que partielle.

Les produits MINERVE de cette étape sont :

- le DOSSIER DES TESTS DES FONCTIONS LOGICIELLES,

- le DOSSIER DE CERTIFICATION DES FONCTIONS LOGICIELLES.

3.4. La validation du logiciel

Les tests conduits pour valider le logiciel sont analogues aux tests fonctionnels mais portent sur le programme complet et sont effectués vis-à-vis des documents de SPECIFICATIONS OPERATIONNELLES DU LOGICIEL et de SPECIFICATIONS DES INTERFACES DU LOGICIEL (produits MINERVE de l'étape de définition opérationnelle du logiciel). Les méthodes et les outils sont identiques à ceux utilisés lors des tests fonctionnels. En particulier, l'utilisation de fonctions opérationnelles validées pour la simulation de l'environnement d'autres fonctions opérationnelles, impose, comme pour les tests fonctionnels, des contraintes de séquencement.

Par ailleurs, les tests de validation sont proches de véritables scénarios opérationnels, ils sont regroupés par "chaînes logicielles" et non plus par fonctions logicielles et les actions sont datées de façon représentative vis à vis du déroulement d'une mission (une chaîne logicielle représente la part incombant au logiciel dans la réalisation d'une fonction opérationnelle).

Enfin, le test est conduit entièrement en boîte noire au niveau des interfaces opérationnelles et les tests de non-régression (tests automatiques) sont utilisés de façon systématique. Ceux-ci élaborent les actions de mise en oeuvre du test et le comportement de référence à partir des informations enregistrées lors du test d'un programme préalablement certifié. Ces informations peuvent être modifiées manuellement, lors de l'exécution du test, pour prendre en compte les évolutions de mise en oeuvre et de contrôle liées à celles du programme testé.

La validation du logiciel constitue une véritable "recette usine" du programme ; celui-ci peut être désormais archivé puis remis au maître d'oeuvre du système d'armes.

Les produits MINERVE de cette étape sont :

- le DOSSIER DES TESTS DES CHAINES LOGICIELLES,

- le DOSSIER DE CERTIFICATION DES CHAINES LOGICIELLES où la conformité des résultats obtenus à ceux prévus est consignée pour chaque test exécuté,

- le SUPPORT MATERIEL et le LISTING du programme archivé,

- le BORDEREAU DE LIVRAISON DU LOGICIEL donnant la référence de tous les produits remis au demandeur.

CHAPITRE 4 : LES OUTILS

Les outils utilisés lors des étapes de test du logiciel sont présentés sur le schéma suivant :

3.3	Tests unitaires	IDAS		
3.4	Tests d'intégration	IDAS	COCODIN	EDL
3.5	Tests fonctionnels	IDAS	BVL	
3.6	Validation du logiciel	IDAS	BVL	

Les outils COCODIN et EDL, spécifiques de l'étape de tests d'intégration, ont été présentés succinctement lors de la description de celle-ci. Les outils IDAS et BVL font l'objet d'une présentation plus détaillée dans ce chapitre.

4.1. L'outil IDAS

IDAS est un système de test de logiciel qui permet d'automatiser le processus de test, y compris dans le domaine temps réel. Le mode de fonctionnement consiste à exécuter le programme testé, observer son comportement et vérifier que ce dernier est conforme au comportement attendu. Le champ d'application d'IDAS est vaste car il peut s'adapter à tout langage de programmation (LTR, Ada, etc) de même qu'à tout calculateur sur lequel le programme testé s'exécute.

A ce titre, IDAS est proposé comme outil de test sur l'atelier ENTREPRISE, atelier de génie logiciel portable multilangage développé à l'initiative de la DGA pour l'ensemble de ses applications.

IDAS repose essentiellement sur l'utilisation d'un langage de test qui permet de formaliser les tests effectués sous forme de programmes de test. Ceux-ci peuvent être archivés et réexécutés, permettant ainsi un "test de non régression" à chaque modification du programme testé.

Ces programmes de test peuvent être compilés ou interprétés. Ils permettent de manipuler, au niveau symbolique, les variables et les instructions du programme testé (test boîte blanche).

A cet effet, IDAS est connecté à la chaîne de production (compilateurs, assembleurs,...) via un post-processeur qui fournit au compilateur et à l'interpréteur d'IDAS les informations qui leur sont nécessaires (adresses, types de données,...).

Le système IDAS peut être utilisé avec un simulateur du calculateur cible. Selon les fonctionnalités de ce simulateur, l'observation temps-réel du programme testé peut être rendue possible ou non. Cette configuration est habituellement utilisée pour les tests unitaires et les tests d'intégration.

La configuration utilisée pour les tests fonctionnels et les test de validation comporte une machine de test et un machine cible distinctes reliées par une interface matérielle qui fournit les capacités d'observation temps-réel (points de contrôle). Cette configuration assure que l'exécution du programme testé n'est pas perturbée. Les points de contrôle spécifient :

- des événements simples survenant lors de l'exécution du programme sous test (exécution d'une instruction, manipulation d'une variable)

- des éléments complexes obtenus par combinaison logique d'événements simples.

Ces points de contrôle ne sont pas, comme il est courant, insérés dans le code, mais sont chargés dans l'interface matérielle.

La structure générale du système IDAS est représentée sur la figure 3.

Le "noyau" du système IDAS est constitué :

- d'un interpréteur qui permet une mise au point rapide des programmes de test,

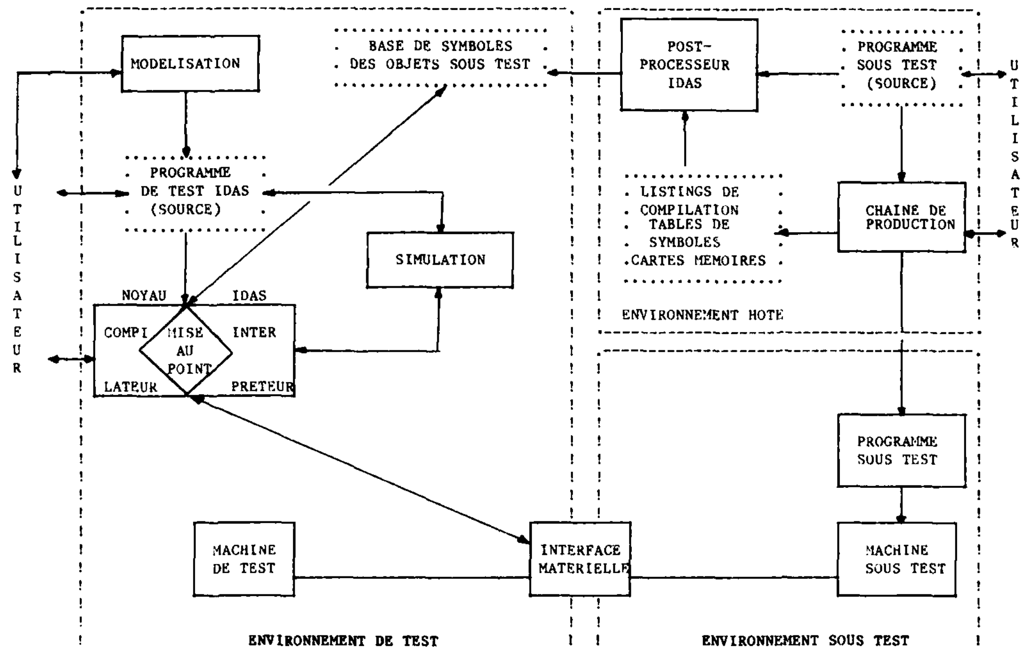
- d'un compilateur utilisé pour exécuter les programmes de test préalablement déverminés à l'aide de l'interpréteur,

- d'un outil de mise au point qui permet de localiser les erreurs détectées lors de l'exécution des programmes de test.

Ce noyau constitue une structure d'accueil pour des outils complémentaires. De tels outils sont déjà réalisés, ce sont :

- un outil de modélisation, qui permet d'exprimer un modèle de comportement en utilisant le formalisme des réseaux de Petri,

- un outil de simulation, qui permet de substituer des programmes de test à des parties de programmes testés non encore disponibles (instructions ou procédures).



ENVIRONNEMENT IDAS

figure 3

4.2. L'outil BVL

La BVL est un ensemble de matériels et de logiciels permettant de tester dynamiquement les logiciels des calculateurs embarqués ESD.

Lors de ces tests, le(s) calculateur(s) embarqué(s) est(sont) connecté(s) à la BVL par ses(leurs) liaisons opérationnelles (bus et liaisons analogiques). Les fonctions réalisées par la BVL consistent en la simulation de l'environnement réel du calculateur, en l'observation du comportement du calculateur embarqué au travers de ses liaisons opérationnelles (test boîte noire) et au contrôle de ce comportement.

La simulation prend en compte une trajectoire avion définie pour les besoins du test et dont les paramètres sont calculés et enregistrés en centre de calcul. L'utilisation d'une console graphique permet de simuler les faces avant des postes de commande et des visualisations opérationnelles. En particulier, elle permet de simuler les actions opérationnelles (actions sur les postes de commande) et les actions de modification de l'environnement (ex : mise en panne d'équipement). Cette console permet également de présenter les informations circulant sur les liaisons opérationnelles et des informations internes aux calculateurs de test et sous test.

Les test sont conduits de façon manuelle ou automatique. En test manuel, la simulation des actions opérationnelles et de modification de l'environnement est réalisée par l'opérateur. En test automatique, cette simulation est issue de l'enregistrement des actions réalisées dans un test antérieur dénommé "test de référence". La liste des paramètres à vérifier et les valeurs de référence utilisées lors d'un test automatique sont également issues d'enregistrements réalisés lors de ce test de référence.

La structure matérielle de la BVL est représentée sur la figure 4.

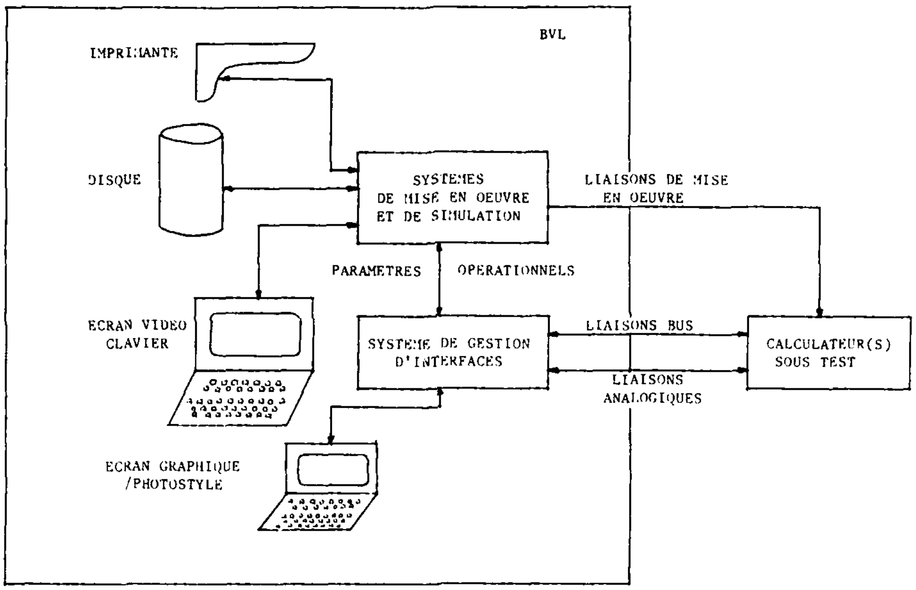


figure 4

La BVL est composée de deux sous-ensembles matériels principaux .

- Un système de mise en oeuvre et de simulation connecté :

  . à la liaison de mise en oeuvre des calculateurs sous-test (arrêt/relance du calculateur),

  . à des périphériques standard : imprimante pour l'édition des anomalies, disque pour les informations nécessaires aux tests automatiques (paramètres de vol, paramètres de référence, résultats de tests), écran vidéo/clavier pour la mise en oeuvre du système.

- Un système de gestion d'interfaces connecté :

  . au(x) calculateur(s) sous test par l'intermédiaire de liaisons opérationnelles (liaisons bus et liaisons analogiques discrets et synchros),

  . à un écran graphique doté d'interfaces souris/photostyle simulant les interfaces opérationnelles.

La structure logicielle de la BVL comporte quatre ensembles principaux :

- un logiciel de génération de paramètres de vol :

  Ce logiciel est exécuté sur un matériel distinct de la BVL (centre de calcul IBM). Il élabore des paramètres de vol cohérents relatifs à des trajectoires avion ; ces paramètres sont transmis à la BVL sous un format compatible avec les échanges bus.

- Un logiciel de contrôle de fonctionnement de la BVL :

  Ce logiciel de mise en oeuvre des fonctions de la BVL présente une interface conversationnelle réalisée sous forme de touches fonctions programmables implémentant des menus arborescents. Parmi les foncitons principales, citons :

  . la mise au point de(s) calculateur(s) sous test au niveau symbolique (hors simulation de l'environnement),

  . les tests de validation/recette (avec simulation de l'environnement) comportant comme sous-fonctions :

    * les modes de mise en oeuvre du(des) calculateur(s) sous test : lancement, arrêt, ralenti, pas à pas (au niveau du cycle),

    * la modification manuelle de l'environnement ("actions"),

    * la sélection du mode de fonctionnement automatique des tests,

    * la sélection des visualisations et des présentations (loupe).

  Ce logiciel est exécuté sur le système de mise en oeuvre et de simulation.

- Un logiciel de simulation de l'environnement :

  Ce logiciel est exécuté par le système de mise en oeuvre et de simulation. Pour chaque équipement émettant des paramètres à destination du(des) calculateur(s) sous test, ce logiciel assure la simulation d'un sous-ensemble fonctionnel permettant de générer ces paramètres.

  Cette simulation est basée sur les spécifications des équipements (spécifications détaillées des fonctions opérationnelles et clauses techniques d'intégration) pour l'aspect fonctionnel et sur les fiches d'interfaces pour l'aspect organique des échanges.

- Un logiciel de tests automatiques :

  Ce logiciel est exécuté dans le système de mise en oeuvre et de simulation.

  Son objectif est double :

  . permettre de recréer des configurations de test. Cette possibilité est intéressante pour l'analyse des cas de panne en test fonctionnel et en validation,

  . automatiser les procédures de tests (mise en oeuvre et observation du comportement) et l'analyse des résultats (comparaison à un comportement de référence). Cette fonctionnalité est particulièrement adaptée à la recette.

CHAPITRE 5 : L'EVOLUTION DES TESTS

L'évolution des tests de logiciel, à court et moyen terme, ne se situe pas tant au plan des méthodes et des outils de test proprement dits qu'à celui de l'intégration du processus de test et des autres activités de développement du logiciel.

C'est donc à l'impact dû à l'apparition d'outils situés plus en amont dans le cycle de vie du logiciel (principalement des outils d'aide à la spécification de logiciel) et à la tendance actuelle à l'intégration des outils au sein d'ateliers de génie logiciel, que ce chapitre sera principalement consacré.

5.1. L'impact de l'évolution des spécifications

Cette évolution se situe à deux niveaux. D'une part la prise de conscience de la croissance exponentielle des coûts des modifications au cours du développement a suscité, depuis quelques années, de nombreux travaux dans le domaine de la définition (et à un moindre degré de la conception) du logiciel.

A l'ESD, cet effort s'est traduit par la réalisation de l'outil DLAO qui permet d'exprimer de façon semi-formelle les spécifications de logiciel. Les concepts de base mis à la disposition de l'utilisateur sont :

- les informations qui représentent les données opérationnelles (ex : mode de fonctionnement du radar)

- les interfaces, support physique des informations (ex : codage utilisé pour la transmission sur un bus multiplexé)

- les événements qui modifient les traitements à effectuer (ex : changement de mode de fonctionnement du radar)

- les états qui caractérisent le fonctionnement du logiciel à un instant donné (ex : mode Air-Sol)

- les traitements qui représentent les tâches élémentaires (ex : calcul de balistique).

Les données ainsi définies par l'utilisateur ainsi que les commentaires associés sont stockés dans une base de données qui est ainsi toujours dépositaire d'une spécification à jour. De nombreuses possibilités d'élaborer une documentation sont offertes, exploitant les possibilités d'interrogation de la base de données.

Une telle formalisation des spécifications doit permettre, dans un futur assez proche, de progresser de façon significative dans l'automatisation des tests. L'ESD mène des travaux dans ce sens, qui visent à automatiser la génération de scénarios de test. L'approche choisie consiste à générer un prototype à partir d'un sous-ensemble de la spécification (défini par l'utilisateur) puis, à partir de stimuli d'entrée, à produire des données de sortie par exécution de ce prototype. Les stimuli d'entrée, qui constituent les données d'entrée des scénarios, peuvent être élaborés manuellement ou générés à partir de la spécification. Dans ce dernier cas, l'utilisateur a la possibilité de limiter les informations intervenant dans les scénarios afin de prévenir le phénomène d'explosion combinatoire. Les données de sortie générées par le prototype constituent les valeurs de référence utilisées lors de l'exécution des scénarios de test.

5.2. L'intégration des outils

La prise de conscience des concepts du Génie Logiciel conduit à une intégration de plus en plus forte des outils. Un exemple significatif est constitué par les travaux entrepris actuellement par le consortium ITI qui regroupe huit sociétés françaises du secteur aérospatial. Ces travaux visent à la réalisation d'ateliers de conception de systèmes avioniques et de réalisation de logiciels embarqués et font l'objet d'une présentation lors de cette conférence (M. SLISSA Aérospatiale et P. LAROCHE-LEVY Avions Marcel Dassault).

De tels ateliers fournissent des mécanismes puissants permettant d'intégrer les outils de tests et les outils de spécification et de conception des étapes symétriques (cf figure 1), selon les principes évoqués au paragraphe 5.1.

Ces mécanismes consistent principalement :

- en une base d'"objets" centralisée, moyen de communicaiton standard entre les outils,

- en un système de gestion de configuration généralisé permettant d'assurer une cohérence permanente entre les objets de la base (spécifications, documents de conception, code, programmes de tests).

Une meilleure intégration sera également assurée entre les outils de tests eux-mêmes (BVL et IDAS) permettant ainsi, à partir du même poste de travail, de piloter simultanément les opérations de tests fonctionnels et de validation réalisées par la BVL et les opérations de mise au point (de pannes liées à des problèmes temps réel notamment) réalisées par IDAS.

CONCLUSION

Notre expérience du test de logiciel avionique nous permet de présenter un bilan largement positif.

L'objectif principal, c'est-à-dire la qualité du logiciel, est atteint : on constate, en vol, moins d'une erreur par an et par programme.

Nous maîtrisons également la tenue de nos coûts et de nos délais, en dépit du fort taux d'évolution de nos logiciels.

Nous constatons également un accroissement continu de notre productivité. Celle-ci s'est accrue dans un rapport voisin de 5 entre 1979 et 1987. Ceci doit être pondéré par l'accroissement simultané de notre consommation en ressources ordinateur, mais le bilan global est positif.

Nos efforts tendent aujourd'hui à maintenir cet accroissement de la productivité, et nous espérons atteindre cet objectif par deux moyens principaux :

a) un effort accru au niveau des tests unitaires et des tests d'intégration car, comme cela a été souligné au chapitre 5, plus tôt les erreurs sont détectées dans le cycle de vie, moins coûteuses elles sont à corriger. En particulier, nous comptons perfectionner les méthodes et les outils nous permettant de mieux maîtriser les taux de couverture de ces tests,

b) par une plus grande intégration de nos outils de développement au sein d'ateliers de génie logiciel (incluant l'utilisation de stations de travail graphiques multi-fenêtres), et leur permanente adaptation aux méthodes de conception des systèmes avioniques.

DISCUSSION

E.Daley, UK

What implementation language is used in flight-critical systems, such as fly-by-wire systems, and what is a typical program size for flight-critical systems?

Author's Reply

We are not in charge of flight-critical systems. These are manufactured by the aircraft manufacturer.

G.Bouche, GE

You addressed software testing in the development/software production phase for a specific subsystem. How do you perform software testing in the maintenance phase at the system level? Is it the responsibility of the user/Air Force? What is the relationship to the original producer?

Author's Reply

Maintenance testing at the system level is not our responsibility. Errors detected during flight tests are first analyzed on the aircraft manufacturer's integration test bench to isolate the failure. Then, the equipment is retested according to the general method described (i.e., MINERVE). This process includes completing different types of forms (e.g., error reports) at different levels.

P.Aouad, CA

Your software validation test is limited to a single box. Do you plan a software validation test for more avionics using an integrated test facility?

Author's Reply

It is not planned. It is performed by a system integrated test facility developed by the aircraft manufacturer.

K.L.Edwards, US

Avionic systems contain more and more software, so system reliability is more and more a function of software. Does your testing enable you to prove quantitatively the reliability of your software, or do you assume your software is perfect at the end of your testing?

Author's Reply

Of course, we cannot assume the software is perfect. No one can. Actually, we have very limited facilities to evaluate the reliability of our software. But, we have some ongoing work on this subject, essentially at the unit test level. This can be done either by an analysis at the source code level or by inserting artificial errors and determining whether they are detected in the testing process.

R.Guiot, FR

Vous êtes spécialiste de l'intelligence artificielle, vous n'en avez pas parlé. Pourquoi?

Réponse d'Auteur

C'est très difficile d'utiliser ces techniques. Cependant, certaines parties de nos programmes commencent à être écrit en PROLOG.

M.Kayton, US

Do the Mirage 2000's mission computers (primary and backup) execute the same software (e.g., perform the same functions)?

Author's Reply

No.

DEVELOPING SYSTEMS USING STATE-OF-THE-ART CAD/CAM TECHNOLOGY

by

Vernon A. Anderson, Code 33181, VHSIC Program Manager
Naval Weapons Center
China Lake, CA 93555-6001
and
LT David J. Brewer, USN, Space and Naval Warfare Systems Command
NAVELEXCEN Charleston, 4600 Marriott Drive
N. Charleston, SC 29418

SUMMARY

STATE-OF-THE-ART CAD/CAM in this paper refers to mission-specific applications with the ability to describe a design that is independent of technology at certain levels. That is, describing a design in a manner independent of any particular implementation, yet allowing integrated verification across design boundaries. Such a high-level approach pays huge dividends in transferring designs in one technology to another, e.g., bipolar gate arrays to GaAs gate arrays. CAD/CAM TECHNOLOGY encompasses more generic aspects of systems development requiring a common base or core of computational tools, applicable across the spectrum of engineering disciplines. We'll present some structures useful for improving understanding of technology, systems, CAD/CAM and for better communications across interfaces. A technology planning and communications framework considering a hierarchical systems schema is introduced for looking at the spectrum of user tasks in the life cycle of electronic products. The Navy's Acquisition Strategy and Plans for Computer-Aided *Design/Computer-Aided Manufacturing (CAD/CAM) are presented.*

INTRODUCTION

The technology planning considerations of Figure 1 give a broad overview of the types of things that are important for successful corporate technology transfer and/or planning. The most critical item implied, but not specifically shown on Figure 1 and part of any technology transfer planning, is optimizing the utilization of time and resources. The industrial world's migration to automation and robotics indicates that buying, installing, and learning how to use new, more cost-effective CAD/CAM tools is part of the answer. At issue are the interfaces for communicating across boundaries in a structured CAD/CAM design process. CAD/CAM technology interfaces are considered to be an element of both common core technologies and mission specific environments depicted in the systems acquisition environment of Figure 2. CAD/CAM implies the use of computers in systems acquisition activities from initial concept, through prototype testing, engineering production, to Fleet introduction and product support. An example process would be computer-aided requirements, specifications, design, testing, engineering, processing, instruction, training, manufacturing, and logistics. Structured definitions to identify and define the interfaces needed for this complex environment start with the Figure 3 system acquisition phases from DOD-STD-2167, Defense System Software Development. Another definition required is for "system" which the *IEEE Standard Dictionary of Electrical and Electronics Terms* defines as:

"(1) an integrated whole even though composed of diverse, interacting, specialized structures and subjunctions, or

(2) an organized collection of men, machines, and methods required to accomplish a specific objective."

The above definition doesn't explain what is meant by the terms "interacting, specialized structures or organized collections" inherent in modern structured design. These are the schemata, the plural of schema for systems. Figure 4 shows a hierarchical schema for electronic systems that extends the definition of a system to include its hierarchy. The analysis hierarchy corresponds to the design or engineering process that covers tasks from concept to product. A hierarchical schema to define electronic systems is necessary by virtue of the way in which electronic assemblies or systems are designed and built. It wouldn't be feasible to support the life cycle activities (describe, design, evaluate, partition, build, etc.) of today's complex avionics systems without this sophisticated hierarchical structure to verify and determine intended and resultant operational characteristics such as response, reliability, fault tolerance, and other behavioral limitations, because in modern environments, physical systems are directly linked to system models via tests, measurements, and diagnostics as indicated in the interfaces depicted in Figure 4. System models are a sensitive information area for the military; they form the basis for calculating performance information used to identify areas of improvement for upgrading systems and to evaluate limitations and measure the most effective way to employ and/or defeat a system.

SATISFYING THE NAVY'S CAD/CAM TECHNOLOGY NEEDS

By reflecting optimization of time and resources to the lowest levels, higher productivity, efficiency, and competitiveness in performing the following identified tasks is needed:

a. Assistance in automating typical tasks such as:
   1) Select, plan, and trade-off methodologies, styles, and tools appropriate for a structured design process.
   2) Build hierarchical models by:
      a) Decomposition and partitioning.
      b) Developing testing criteria by defining hardware, software, and associated test vectors..

* TECHNOLOGY PRESENTLY AVAILABLE.........NOW (FOR CRITICAL EXPERIMENTS)

* TECHNOLOGY AVAILABLE FOR PRODUCT IMPROVEMENTS......LOW RISK 1-2 YEARS
HIGH RISK 3-5 YEARS

* ADVANCED TECHNOLOGY
FOR PRE-PLANNED PRODUCT IMPROVEMENTS.........LOW RISK 5-7 YEARS

TECHNOLOGY TIME FRAMES

---

* RESEARCH

* DEVELOPMENT (EVALUATION) AND TECHNOLOGY TRANSFER
a. INFORMATION GATHERING AND DISSEMINATION
b. TRAINING AND ASSISTANCE
c. PHYSICAL, FINANCIAL, AND HUMAN RESOURCES

TECHNOLOGY INITIATIVES AND PHASES

---

* LOGICAL
STRONG LOCAL RESEARCH EFFORTS
SKILLED POOLS OF LABOR
AVAILABLE FINANCING
PRESENCE OF CORPORATE HEADQUARTERS
TRANSPORTATION
CLIMATE AND CULTURAL AMENITIES

* OTHER (PLANNING is worthless without a strategic vision and goals)
IDENTIFICATION AND FOCUS ON NEEDS/REQUIREMENTS
anticipate the future and reflect to the lowest levels
ADAPTATION TO INTERNAL AND EXTERNAL CONSTRAINTS
LINKAGE WITH BROADER DEVELOPMENTS (options succeed, either/or fails)
LOCAL INITIATIVE AND PARTNERSHIP WHICH WILL DEVELOP SUPPORT NETWORKS
and bring local representatives into the design and
implementation of the initiatives
SUSTAINED EFFORTS and GROWTH via coordinated individual efforts

FACTORS AFFECTING SUCCESSFUL DEVELOPMENT

BETTER WEAPONS........FASTER DESIGN CYCLE
USABILITY, AVAILABILITY, AFFORDABILITY, PERFORMANCE

a. & b. TRAINING

b. & c. TOOLS

c. COMPONENTS/PROJECTS

COMPONENTS AFFECTING SUCCESSFUL DEVELOPMENT

FIGURE 1. TECHNOLOGY PLANNING CONSIDERATIONS

c) Support structured hardware and software codesign to allow implementation, performance, and interfaces to be identified so they can be evaluated.

3) Verify subsections of a larger system at the levels of analysis shown in Figure 4 by evaluating simulated test and performance measurements of a design against requirements and constraints.
   a) Validate that the design meets allocated or predicted reliability and diagnostics (testability) requirements.
   b) Validate the physical design prior to committing to fabrication and procurement via sophisticated models and simulations.
4) Extract data.
5) Layout and simulate.
6) Establish a documentation base line.

b. Capability for top-down design and assessment.
1) Allow designers to work at a higher level in the Figure 4 hierarchy to reduce design errors and times.
2) Each user-created object or model may have a minimum structure and hierarchy of:
   a) Phase (see Figure 3).
   b) Level (see Figure 4).
   c) Version.
   d) Date.
   e) Information categories of Figure 3 associated with levels of Figure 4.

```
                 MISSION-SPECIFIC ENVIRONMENTS
      MULTI-SYSTEMS                          BOARDS
        SYSTEMS    <=>                   <=> DEVICES
        SUBSYSTEMS        CORE                CHIPS
          MODULES      TECHNOLOGY             CELLS
                       ENVIRONMENT
\INFORMATION \     TO SUPPORT NAVY ACQUISITION PROGRAMS
\ACQUISITION \         AND SYSTEM LIFE CYCLE PHASES

 INFORMATION | Concept   |              |            |Production
 CATEGORIES  |Exploration|Demonstration |            | --->Deployment
             |           |     and      |Full-Scale  | ---->Support
             |           |  Validation  |Development |   --->Redesign
 TECHNICAL   |           |              |            |
 NON-TECHNICAL|          |              |            |
```

FIGURE 2. SYSTEMS ACQUISITION ENVIRONMENT

\INFORMATION \ SYSTEM LIFE CYCLE PHASES

	ELECTRONIC DESIGN DATABASE CATEGORIES	Concept Exploration	Demonstration and Validation	Full-Scale Development	Production --->Deployment ---->Support --->Redesign
a.	Connectivity	1	2	3	4
b.	State	1	2	3	4
c.	Models	1	2	3	4
d.	Relations	1	2	3	4
e.	Quality Assurance	1	2	3	4
f.	Materials/ Processes	1	2	3	4
g	Testing/ Testability	1	2	3	4

Classes of models in the system life-cycle model correspond to the numbers in the table below. The classes are defined as good enough for:

1) Research and concept exploration.
2) Schematic and limited analysis to build a prototype that will function correctly.
3) Analysis in the hierarchy of Figure 4.
4) Safe use, reliable operation, and affordable maintenance of the product by the Fleet.

FIGURE 3. SYSTEMS ACQUISITION PHASES

c. Capability for bottom-up implementation:
   1) Determine consistency and completeness of information transmitted between groups working on a partitioned design or between levels of a design:
      a) Requirements.
      b) Specifications.
      c) Functional performance measurements.
      d) Functional performance constraints.
      e) Algorithms, test vectors, and diagrams.
      f) Hardware/software architectures.
      g) Information categories of Figure 3 and 4.
   2) Verify the design as it progresses incrementally from one level to the next in the design hierarchy; or from one phase to the next in the life cycle through:
      a) Control and verification of the configuration to reduce errors throughout the design cycle.
      b) Design database/documentation detailing the original design and its history to facilitate refabrication, redesign, and maintenance of the configuration.
      c) Incremental verification of the completed design against the design database elements of Figures 2 and 3, and the levels of Figure 4 by controls and measurements of what:
         1. Checks have been done.
         2. Constraints have been met.

	SYSTEMS HIERARCHY			ANALYSIS HIERARCHY
	INFORMATION/ DATABASE LEVELS	HARDWARE LEVELS	SOFTWARE LEVELS	ANALYSIS LEVELS
a.	Schemata	Multi-Systems	Environments	Behavior
b.	Libraries	Systems	Operating Systems	Algorithm
c.	Encyclopedias	Subsystems	High-Level Languages	Architecture
d.	Dictionaries	PC Boards Modules, Hybrids	Assembly Languages Macrocodes	Block/ Function
e.	Books/Records Families	Discrete Devices Packaged ICs	Microcodes	Logic/ Circuit
f.	Fields	Chips Macrocells Registers	Fields	Structure/ Geometry
g.	Cells	Gates Cells	Bits	Procedure/ Process

	INTERFACES			
h.	Translators/ Formats, i.e., Neutral file, Intermediate	Channels Bus Interfaces, i.e., Optical or VHSIC PI Bus	Compilers Reverse Compilers	Diagnostics/ Tests/ Measurements

1) Information translators will be based on:
   a) IGES/PDES for graphic information.
   b) VHDL and EDIF for non-graphic information.

FIGURE 4. INTERFACES OF THE ELECTRONIC SYSTEMS AND ANALYSIS HIERARCHIES

CAD/CAM/CAE IMPLEMENTATION PROBLEMS, DIFFERENT LEVELS, DIFFERENT NEEDS

Not only are modern electronic designs impossible to accomplish without design automation tools but engineers in government and industry are faced with ever increasing design times because of increased complexity caused by miniaturization. For example, approximately 5 Gbytes of design data are required to describe a 300K-transistor device such as Motorola's 68030 microprocessor. The number of primitives we can put on a silicon chip is presently growing at an exponential rate. Integrated circuits containing over two million transistors are on the drawing board; their macros are under fabrication. These devices are among the most complex things man has built with large teams of researchers working on their eventual realization. Not only are very large systems continuing to shrink, but a complexity corresponding to neural network structures of the brain is rapidly being approached. These devices are multi-systems and are every bit as complex as the number of devices they contain implies. Thus, it is painfully obvious to design engineers at the lower levels that complex design, development, and product integration may only be accomplished within a reasonable time by off-loading the mundane, time-consuming, "stupid" tasks to computers. This is at a lower level than Artificial Intelligence (AI) activities which are focused on the next level up of adding more intelligence to computer tools. The thrust of this next level up is to off-load tasks such as design or electrical rule checks that require some intelligence buried in the design tool or database so that the computer may act as a design assistant. The next step-up from this is to specify open architecture CAD/CAM/CAE systems that will allow for the integration of multiple vendor tools with an eye toward full implementation of AI tools.

SATISFY FRONT-END ACTIVITY NEEDS FIRST, FOR THE HIGHEST PAYOFF

The Computer-aided software engineering environment has developed data which shows that the earlier errors are found, the cheaper it is to correct them; the cost of correcting errors grows rapidly with time. This principle requires designs be "run through the mill" as early as possible for their evaluation. The lesson here is to support a thorough understanding and evaluation of the system development or life cycle phases shown in Figure 4, and target maximum investment in the early phases of development for maximum payoff.

THE U.S. NAVY'S APPROACH TO CAD/CAM DESIGN AUTOMATION

The Navy has been implementing a three-phased strategy to provide its workers with state-of-the-art, cost-effective CAD/CAM tools to achieve a better-engineered, more maintainable product than ever before and to satisfy its front-end needs.

PHASE ONE—THE FIRST ACQUISITION

The Navy embarked on the first phase of its design automation strategy in 1981 with a Navy-wide centralized procurement known as CAEDOS (Computer Aided Engineering Documentation System). This acquisition was a 99.9 million dollar contract from 1981 to run through the middle of 1988 and provided a minimal number of CAE systems to a small portion of the Navy. The major problems encountered in this introduction of design automation in 1981 were not technical, but were related to the nature of people; they were asked to accept and embrace something new. This new technology was sophisticated, complex, and difficult to use without extensive training. It was a new way to do a job and required technical guidance, extensive training and motivation on the part of the users to get involved in doing their jobs better by making use of this advanced technology. It was embraced by those who had a desire to do something new and better. Whether or not it was successful depends on your point of view. The CAEDOS acquisition demonstrated the viability of a centralized contract vehicle for activities to buy state-of-the-art technology at significant price reductions but it didn't solve the problem of what, how and where to obtain technical guidance in effective application of new technology. Technology education and acceptance are separate problems. The problems are the same ones American education has today, how to effectively train and motivate people to do their best. There are indications that some CAEDOS assets are falling into disuse today due in most part to the fact they have been surpassed by technology which is cheap to obtain and much easier to use. In isolated cases the lack of economies of production have caused these assets to fall into disuse. Adequate individual training and technical guidance in applications is a continuing problem which is being addressed in the second acquisition specification by requiring built in features to address these needs.

PHASE TWO—THE SECOND ACQUISITION

The Navy is presently looking at benchmarks for evaluating what to procure for its CAD/CAM second acquisition. This second acquisition is an ambitious effort which has a user developed specification that was released by the Navy to industry early in 1986. The proposed systems are to provide cost effective, commercially available, non-developmental, tools to:

a. Shorten the design cycle.
b. Promote better, less expensive designs:
   1) Build in reliability, maintainability, and availability of equipment in the design process itself: structured design is a feature of the tool set.
   2) Manage configuration over a product's life cycle for early design capture with self documenting and instruction features built into the tools to attract and train users in a natural, friendly environment.

The hardware and proposed software tool sets in the specification are vertically integrated within five award bands defined by NAVSEA, NAVAIR, NAVSUP, NAVFAC, and SPAWAR. The specification identifies and defines the hardware components shown in Figure 5, linked through a local area network also defined in the specification. An open architecture analogous to Figure 2, consisting of core functions common to the different Navy Systems Commands with mission specific-software on top of this core, is being stressed.

Once the technical hurdle of identifying the systems to do the defined tasks has been accomplished, some figure of merit to measure the cost benefits must be "bottom-lined." Let's compare a production-oriented private corporation against a design-oriented government agency to investigate the development of a quantifiable figure-of-merit with some common ground in this spectrum.

A private company derives economies of production in an integrated design, manufacturing, marketing and service environment. The design engineers in the private sector are more likely to be tightly coupled with the production engineers, salesmen, and service technicians and the desired result is a producible more maintainable design; competition forces industry to "design to production and customer satisfaction." The mass-produced component rolls off the assembly line with quantifiable regularity or the team established to maintain the product line doesn't have a job.

In contrast is the design that never appears (with emphasis on appears) to go anywhere. The output is nothing more than a design database which describes a design and is commonly called a drawing package. Drawing packages are classified into levels roughly corresponding to their completeness (system phases) and/or verified accuracy; level 3 drawings follow many standard conventions and invariably require the services of a competent draftsman who understands these standards. The element of enlightenment that often escapes the reviewer of a design database or drawing package is the technology required to get that "intermediate" output. Yes, a design database (commonly a digital database) is nothing more than an intermediate form—as yet an non-quantifiable product. It is the culmination of the research, engineering prototype, or engineering development phase required to describe the product to be produced. In many firms and government agencies this intermediate form has been the end-product. In firms specializing in design database development, obsolete CAD/CAM/CAE systems become nothing more than 2-D automated drafting tools to offset the costs associated with their purchase. Productivity has been measured on near-term investment return rather than against a more reliable, maintainable, or available design which can be verified through sophisticated simulations of the product before funds are committed to production.

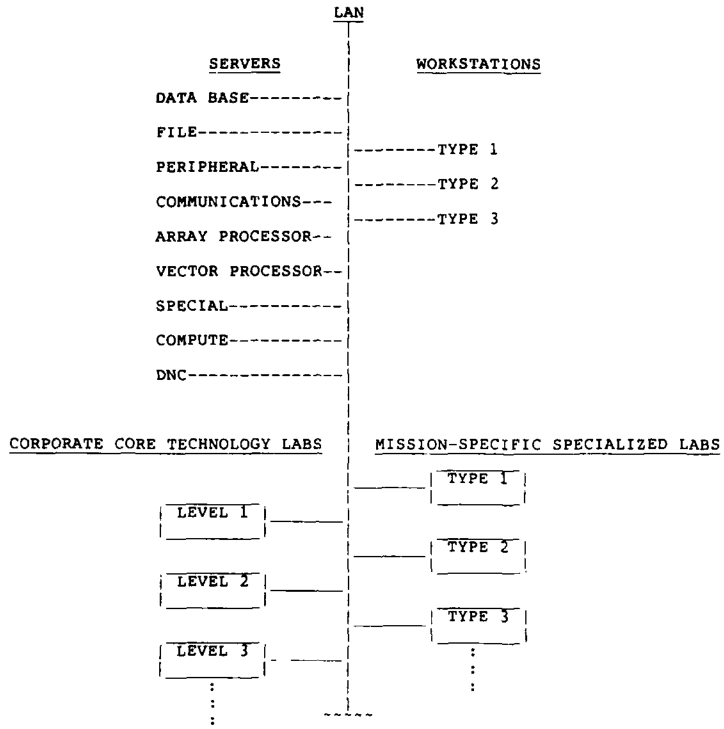


FIGURE 5. OPEN ARCHITECTURE OF THE NAVY's CAD/CAM SECOND ACQUISITION

TRANSFER TECHNOLOGY GRACEFULLY

The Second Acquisition plans to provide hardware and software deliverables which will be the leading edge of technology. The specification looks into the near future and reflects this future to the needs at the lowest levels of design tasks. The document is unparalleled in private industry because it provides a cohesive view of the direction automation is heading from diverse viewpoints. It calls for a full spectrum of vertically integrated design tools in mission spectra which correspond to the Navy's Systems Commands yet has a common core upon which to build these diverse applications. The inputs from industry were used to help scale the document to reality so that the CAD/CAM systems to be benchmarked in 1987 will in fact be the yet-to-be-announced products scheduled for release in 1988 or the leading edge products.

These leading edge tools must be phased into tasks for which they are appropriate and will give the highest payoff. There will be problems of introduction and careful nurturing of the user base will be necessary during their infancy in 1988. The CAD/CAM tools will be in danger of falling into disuse as a result of lack of understanding their power or full-function capabilities, particularly in reducing costs of designing to production constraints. History has shown that centralized procurements fail when the user base is not represented. Fragmented procurements result in a fragmentation of common database information and a fragmentation of design principles. This exacerbates "islands of information" problems even within single activities. A structured approach must be pursued to allow a standardization of communications and design principles resulting in maximum utilization of shared resources, including peripherals and design data.

FUNNELING TECHNOLOGY

There are two steps to the approach to injecting CAD/CAM technology within the context of the Second Acquisition. The objective of the first step is to let a separate contract award for each Systems Command as a part of the Second Acquisition. To perform this task in a coordinated fashion requires a dedicated Technology Office within each Systems Command to address technology issues specific to the different missions.

The objective of the second step of the Second Acquisition is to provide what is called "technology refresh" through centralized guidance and expertise. "Refresh" addresses evaluating the usefulness, appropriateness and availability of CAD/CAM technology as it evolves in private industry and is ready to be used to solve Navy problems applicable to its large user base. This requires in-depth knowledge of design efforts throughout the Navy, with particular attention being paid to the methodologies and design techniques being applied in field agencies, and a technology group attuned to evolving requirements and solving real problems. Mission-specific Technology Offices formed to address Second Acquisition CAD/CAM application problems will become well-versed in CAD/CAM procurement and implementation techniques and will be capable of injection of this technology into their own organizations through their relationship and association with private industry, other acquisitions, and other Technology Offices. The participants within each office must be classified as technologists and must be fully cognizant of current research, prototype and engineering development and manufacturing efforts at the various Navy agencies in order to react dynamically to user requirements. Office personnel will funnel evolving user requirements to industry in an effort to furnish vendors advanced information on where to focus their advanced products. Only through this type of dynamic information interchange will technology refresh make sense in the context of the Second Acquisition. Serving as a repository of technical information on Navy CAD/CAM systems, the Technology Offices will be looked upon as invaluable resources that will champion strong cases made by the user base and funnel industry standards to these same users in a structured manner. The full gamut from design to product can be addressed through this highly integrated and structured approach.

PHASE THREE—THE THIRD ACQUISITION

The Third Acquisition will take the Navy into the next century, providing a fully integrated CAD/CAM/CAE system for Navy scientists, engineers, and other personnel. The third phase for full horizontal and vertical integration is slated for beyond 1992 of this multi-year, three-phase effort. Total integration means a single computer-aided environment/schemata for accomplishing concept to product and support in all the engineering disciplines. The goal is to have total or maximally optimized, integrated systems across technical life-cycle applications in the Navy to fully integrate both vertically and horizontally, linking and cross-linking processes, thereby establishing a centrally controlled, cradle-to-grave warfare system model with distributed simulations of system coomponents. This concept will evolve through the efforts of technology planning at all levels to provide migration paths for government, academia, and industry efforts. Development of the concepts and standards needed for evolution to and implementation of the Third Acquisition will only be attained by close cooperation and relationships on an international scale.

Interfacing and Integrating Hardware and Software Design Systems

Daniel Davis, Associate Professor
Department of Computer Science
Naval Postgraduate School
Monterey, Ca. 93943

Summary

Fundamental problems in interfacing and integrating information between diverse design systems are examined. Specifically problems associated to the contexts of meaning that are required to understand information in diverse systems are examined. Increased research into a general theory of design is advocated. The resolution to some integration problems is suggested on the basis of recent developments and experience with a functional design theory.

1. Introduction.

There is now considerable technological movement towards the use of automated design systems in the design of mechanical components of systems, in the design of VLSI components, and in the design of software systems. Each of these activities have proceeded in parallel, without a great deal of coordination and cross fertilization of technologies, with the result that it is becoming increasingly difficult to establish interfaces between these systems or develop an integrated approach to the entire system design problem.

At the same time, there are a number of ideas and principles common to the design systems being developed for different technologies. Most of these systems have a means of specifying the requirements of the design, all have some means of describing the components of a design, and all have some means of describing designs in terms of components. Some systems use the technology of expert systems to aid in the design process. There are a variety of user interfaces to the information in these systems, so-called views. Some views are oriented to the designer, some to the system design management process, and some to the manufacturing or implementation process. The requirement for multiple views or multiple interfaces within a single system is being generally acknowledged as an important concept.

Yet the possibilities of developing and integrating these diverse systems into a single system still seems remote, and fortunately, may be undesirable at the present time. Close coupling between diverse technologies could seriously impede progress in any one of them. At the same time, there is a need for some coupling. There is a classical tradeoff here. Should a system be integrated so that it optimizes the interaction of information among all those who know it, but forces the user to express himself in limited ways, or should there be a variety of expressions, and thus increase expressivity of individual environments, but correspondingly decrease the communication between them?

A middle solution lies in providing means for standardizing the interface methodology between a variety of systems. As a first step, it would be an improvement if we could develop a means of uniformly relating information in one type of environment to information in other environments. This would provide a means of transmitting information from one environment to another in a systematic manner. The problem of doing this is similar to the problem of having a standard internal representation of information, for which multiple views are possible, but which establishes a consistency between views, in a single system. These problems arise equally in mechanical, VLSI, or software design systems. It is the information capture problem.

There is another issue here. Up to the present time, system design, in each of the technological areas in which it is expressed, has not had a theoretical basis. It is in the same state as other disciplines in an early stage; a collection of techniques, tools, and unformulated processes in the minds of users. This will probably remain an essential fact in system design for some time to come. However, the demands of the problems that need to be solved, and the need for capitalizing on previous efforts create a need for formulating established principles and techniques. Such a process is one toward greater generality and abstraction, and ultimately leads to a better understanding of the processes themselves.

The primary objective of this paper is to describe a limited methodology for describing system functional design

information independently of the technology used to realize the design. That is, in terms of 'what', rather than 'how'. This methodology can be used to establish interfaces between various design environments by establishing a way of describing design information independently of its technological idiosyncrasies. In some sense, it provides a methodology for describing a view of a design component in terms of its interface to the outside, without regard to how its properties are achieved. Equivalently, it provides a means for modeling and manipulating design elements abstractly. Yet it is also general enough and precise enough to describe specific technological properties of design elements when required. An important feature of this methodology is that it has a rigorous theoretical basis. This is a feature we feel is lacking in many existing approaches to design. Also this paper is a report on work in progress toward the goal of achieving true 'resource abstraction'; the goal of describing and designing systems in formal terms, so that design problems can be solved in principle, not just in terms of a current technology.

In the following, we establish the background of issues and ideas, then introduce the general principle of functional abstraction , and then a specific theory of resource abstraction. Next we illustrate these ideas with a simple example, and conclude with some remarks on future directions.

2. Background.

The primary reason that it is difficult to develop interfaces between different design systems is for the same reason that it is difficult to develop interfaces between any two computing systems. There are few standard methods for describing general information. Even the simplest types of information, such as a yes/no answer to a question has no consistent representation in a computer. We must begin with this fundamental problem if we intend to establish interfaces between a variety of systems.

In spite of the fact that there are ISO and IEEE efforts to standardize many aspects of the design process, such as ADA, VHDL (VHSIC Hardware Description Language), EDIC (Electronic Design Interface Format), IGES (Initial Graphics Exchange Standard) and PDES (Product Description Exchange Standard), it is neither necessary, nor is it practical, to expect that universal standards could be formulated and enforced across a wide spectrum of design activities. Although standards can be imposed successfully witihin a limited context, experience has shown that broad standards are generally unenforceable and tend to constrain the free development of technologies and ideas. System design is in an early stage of development, and it would be premature to fix on immature concepts or principles. On the other hand, total chaos is also counterproductive to our ability to capitalize on knowledge in one domain by applying a similar knowledge in other domains. This is the problem that is already manifesting itself today. The solution is to find some reasonable middle ground. This middle ground must be based on something that is fundamental to all design systems. We need to view the processes of design in abstract terms and synthesize elements of the process as practiced in its various contexts.

All design systems presuppose a collection of elements that are combined to form a design. A design establishes a juxtaposition of and interrelations between these elemental components to achieve a desired result. Design is inherently hiearchical, in that the result of design at one level can become elemental components at another level. The properties of the design follow in some complex way from the properties of the elemental components and the way that these components are combined. The hierarchical nature of design also reflects the fact that our primary strategy for handling complexity is divide and conquer.

In some cases, design elements are physical abstractions that represent some material object. In other cases they are functional abstractions that represent constructs from mathematics or a field of engineering. Some of the aspects of the process of design represent temporal properties of the sytem as opposed to static properties. In fact, it is a safe generalization to say that all design systems have physical, functional, and temporal dimensions. A design consists in elaborating a hierarchy of components that are interrelated in each of these dimensions and combine to create a complex resource. At our present state of knowledge, a unified theory that can handle all these dimensions is too much to expect.

In practical situations, there is usually a great deal of information about design components that is assumed by the designer as part of general knowledge of a technology. This information about components is not expressed in the system, and may be only vaguely understood by its users. In most cases, users must expend a considerable effort just to learn the context necessary to use a particular system. This is one fact that makes the interfacing and integration of design systems so difficult. In order to pass information from one system to another, or even one part of a system to another part, we have to have some confidence that the information is meaningful. This is the 'semantic gap' problem. There is a gap between what a design component is in the system and what it is in the mind of a user. At the current state of the art, the designer bears the burden for this discrepency and may have little or no automated assistance. Even if the system has means for capturing the meaning of a description in the system, such as some rule

based reasoning, it is still difficult to export this information to another system because its meaning requires the proper context. We can illustrate the problem with a simple analogy.

Suppose I type an English language definition of a design into an editor using ASCII characters. In this case, the design information is encoded into bits in a file. Not only is there a great deal of information that this design may express that is not in the file, unless I know that the data is expressed in ASCII, even its meaning as characters cannot be communicated. If I did not know it was in English, another essential context for its meaning would be missing also. Stated more generally, data by itself is always meaningless. It is data within a context that has meaning. In the functional perspective, the functions that can be applied to the data determine to large extent the meaning of the data.

A simplistic solution to the problem of exchanging information is to establish a universal standard for the information; effectively create a universal context of meaning. In the above example, if there is a universal standard that tells me that data is in ASCII, I can get at the information it contains. If I know that the character information is VHDL or ADA, I can get more information out of it, since I can apply a larger context of meaning to it, if this context is known to me. However as we remarked before, this seems to be a hopeless, and even undesirable, approach in the case of design systems. Much of the context of meaning of a particular design environment is known only to users familiar with it. A more modest approach is to provide a methodology for establishing, along with the information, the context of meaning required to understand it. In very practical form and in a limited manner, this concept has already proven its use on the Macintosh system I am using to type this paper. First the system assumes some widely known context of meaning necessary to perform an activity. For example typing text on a regular typewriter is such an activity. Such a known activity is called a metaphor. Then the system attempts to recreate that activity in a manner consistent with the metaphor. The environment required to display the character data and manipulate it, that is, the aggregate of functions used to edit and display the characters in these sentences, creates the metaphor and is stored with the textual data. Whenever the textual data is loaded, its context is assembled and also loaded. The aggregate of functions used to manipulate the data, to create its context, is called its resources. The resources establish the meaning of the data. More precisely, the resources establish a bridge to the context of meaning. It is the data within the context of the functions that has meaning. This is a simple, yet powerful concept. In the following sections we will describe a similar approach to the description of design information that seems broadly applicable to the functional dimension of design.

3. Abstract Functional Resources

There is a growing body of research in the areas of programming linguistics and software engineering, that suggests a method for resolving some of the difficulties discussed above (Refs. 1, 2, 3, 4, 5). This research has grown out of efforts to design components within programming languages, and subsequently, within general software systems. It is usually called the theory of abstract data types, although at the present time, there is no single theory but rather a methodology that is evolving to handle design complexity in software systems.

There has been a realization in software design that we are continually re-implementing the same functional resources within a variety of environments (Ref. 6.) For example, the part of a software system that accesses files stored on a disk always includes functions to create, open, close, read, and write files. Software systems may perform these functions in a variety of ways, but in some abstract sense, they all perform the same functions. Substantial portions of software system design consists in defining a hierarchy of functional interfaces of greater and greater abstraction. At any level it is aggregates of functions thata define the resource. For example, the file functions above are useful only as an aggregate. Any one, without the others, is meaningless. We will call such a functional aggregate, together with the types of data it manipulates, a resource. Resources are the elements of functional design. In software system design, we would like to have a way to precisely describe resources abstractly, without regard to the specific programming language, operating system, or hardware used to realize them. We also wish to establish the ways in which resources combine to create more complex resources. We would like to describe the essence of these operations in a manner that allows us to reason about them consistently. This would be a step toward a true design *theory for the functional dimension of systems design. The primary problem to be solved is how to associate meaning* to a formal description of a resource. How do we insure that every user of the design specification interprets its meaning in the same way? Also, if we wish to aid the designer with automated tools of reasoning, we must be sure that the tools manipulate the specification in meaningful ways. This is the problem of specification or description semantics. These same problems arise in any attempt to develop an automated design system, and in any attempt to interface or integrate systems. It seems that the greatest progress in this direction is associated to the theory of abstract data types.

There are several features in this theory that may contribute to the general problem of integrating and interfacing design systems:

Firstly, the approach begins with an attempt to solve the semantic problem for design components. Not only is there a discipline of how to go about designing components, there are theories associated to the process of description that provide rigorous meaning to design components. This means it is possible to automate much of the reasoning about components in a way that is faithful to the model in the mind of the designer.

Secondly, the methodology seems to be very general. Although, it started out as a method to be used for software functional design and has its origins in programming linguistics, experiments indicate it can be used for the functional design of hardware.(Ref. 7, 8, 9,). Since it has crossed these boundaries, it holds some promise as a basis for interfacing between them.

Thirdly, it is an abstract methodology. It naturally provides a method of design that is independent of particular technologies of implementation. In software, this means that the language of implementation is not a necessary issue during design, and for hardware, it means that the manufacturing technology is not a necessary issue of design. At the same time, it allows issues of implementation to be made issues of design, to any reasonable degree. In practice the methodology reflects the fact that abstract objects are simpler and more rarefied and thus easier to define, while objects used for implementation are more complex, and therefore more difficult to define. Hence, if a methodology is used that forces us to define objects precisely, there will be a bias towards defining them in their most essential, abstract, and thus simplest terms.

At the present time, the theoretical work on abstract data types has been focused on describing components strictly in terms of the functions they perform. There has been less work on describing the temporal properties of system components. The physical properties of components, such as size, power reqirements, material properties, etc. have yet to be addressed by this theory.

This theory is hierarchical in nature. Design components can be described in terms of more primitive components. Fundamentally, a resource is described by listing the names of the operand types it requires, and the operations (functions) that are available in the resource to manipulate values of these types. As we have indicated, the description is a purely functional one. Each operation has a fixed number of operands, with fixed types, and a fixed return type. The operations generally include functions that generate the primary values of each operand type (constant functions). This description of the resource defines the primary functional interface to the resource. Additional parts of a resource specification describe the properties of the resource, and ultimately define the actual meaning of the resource by application of an associated semantic theory. In other words, the semantic theory can be used to systematically determine which 'real' objects provide implementations of the resource defined in the specification. Error conditions can also be defined in the specification of properties. Although much of the flavor of this approach is influenced by its development from software design, since software is fundamentally abstract , the method is also suited for expressing general functional resource notions.

4. Resource Specifications

In the following, we will take a brief look at the form of the resource specification methodology. A more complete treatment can be found in the literature (e. g. Refs. 1, 3, 4, 10). The actual syntax for specifications is not standardized, although most variations are relatively minor.

Resource objects are described in two parts. The first part is the syntactic part, which is used to describe a form that the resource can take. The second part, the semantic part, is implied by the syntactic part.

A form for the syntactic part is:

Resource [Resource name] is:

Operand Types:
[Operand type set]

Operators:
[Operator specification set]

Properties:
[Property specification set]

The operand set is simply an unordered list of names for the operand types of this resource. The operator specification set is an unordered list of the names of the operators together with a description of the number and types of their operands and result. The property specification list is a list of properties satisfied by the operators.

For the purposes of illustration, consider the following example of a resource specification for the yes/no resource.

Resource Boolean **is:**

Operand Types:
Bool

Operators:
True: -> Bool,
False: -> Bool
Not: Bool -> Bool
And: Bool, Bool -> Bool

Properties:
Not(True()) = False(),
Not(Not(b)) = b,
And(True(), b) = b,
And(False(),b) = False()

Note that constants, such as True, and False above, are represented as nullary operators (or equivalently, constant functions).

The fundamental idea of a resource specification is not that the actual resource has the exact form given in the specification, but that in essence, it achieves an equivalent functionality. This is stated precisely in the accompanying semantics of specifications, i.e., what is meant by the above syntax. Without going into the semantics in detail, let us just remark that it is a consequence of the semantic theory associated to such specifications, that any renaming of the above operands or operators above, or any replacement of the operators by equivalent primitives, such as 'not' and 'implies', or any replacement of the axioms by other mathematically equivalent ones, would have the same meaning. Thus the specification above is only one of many descriptions that denotes the same abstract resource. And equally, the abstract resource has many equivalent realizations, or implementations. In this sense then, the resource description is a way of capturing the 'abstract' essence of the resource.

Boolean is a primitive resource. It is not built up of more primitive elements. It is atomic. Specifications can also be built up hierarchically. To illustrate, consider a functional specification for a stack resource that can hold boolean values:

Resource Boolean_Stack **is:**

Extend Boolean **with:**

Operands:
Stk, Bool.

Operators:
Mtstk: -> Stk
Push: Bool, Stk -> Stk,
Pop: Stk -> Stk,
Top: Stk -> Bool,

Properties:
Top(Push(b, s)) = b
Pop(Push(b, s)) = s

Undefined(Top(Mtstk()))
Undefined(Pop(Mtstk()))

The extend directive can be interpreted as: "include and expand the named specification to include the following operands, operators, and properties." Of course, when a resource is included as part of a larger resource, in the context of the larger resource, its properties could change. In the above example, without an examination of the properties of Boolean_Stack, we can't be absolutely sure that we haven't altered the properties of Boolean, as given in its specification. (In fact, in this example, it isn't difficult to check.) A resource whose properties remain unchanged within a larger context is said to be 'persistent' in that context. The semantics associated to specifications provides a systematic way to check for persistence and exemplifies, in one instance, the importance of rigor.

Note that whenever a resource is specified operations in addition to the given ones are implied by combinations of the given ones. For example, considering only the Boolean resource, we can form expressions such as:

And(Not(True()), Not(And(True(), False())))

In general, an infinite number of such expressions are possible. We call such combinations 'formal expressions', or 'formal terms'. A formal grammar can be written for these formal expressions as follows:

```
<BoolTerm> ->        True() |
                     False() |
                     Not(<BoolTerm>)|
                     And(<BoolTerm>,<BoolTerm>)
```

Similarly, the Stack resource allows us to create more expressions in addition to these using the expressions from the Boolean resource. For example, if $v_1, v_2, \ldots v_n$ are Boolean expressions, then the following expressions illustrate formal terms from the Stack resource.

Push(v_6,Push(v_3,Pop(Push(v_1,Push(v_5, Mtstk())))))
Push(Top(Push(v_1,Push(v_3,Mtstk()))), Push(v_2,Mtstk())

A formal LL(1) grammar can be written for these additional terms by extending the above grammar as follows:

```
<Term> ->        <BoolTerm> |
                 <StkTerm>

<BoolTerm> ->    Top( <Stk Term> )

<Stk Term> ->    Mtstk() |
                 Push(< BoolTerm> , <Stk Term>) |
                 Pop(<Stk Term>)
```

Thus the problem of recognizing and generating all correct formal terms given an arbitrary specification is straightforward. Note we require that there is at least one constant term (nullary term) of each operand type. For example True() and Mtstk() are the constants for Bool and Stk. Note also how formal expressions are created on top of the expressions that we already have and that there are formal terms for each type of operand.

Just as the operator list is a finite list from which we can create an infinite number of operations, the property list is a finite list from which we can induce an infinite number of properties, which are in fact, relations between operations. Properties effectively specify which operations have an equivalent result. The precise rules for deducing such relations between formal terms are determined by the semantics. Although, this may seem to be a simple matter, it is a more difficult problem than it appears.

In Refs 7, 8, 9, 10 these methods for describing and specifying system components were applied to the design of an abstract processor, including data values, memory, and instructions. Subsequently, the same methods were applied to the design of an abstract display device, and database. All these specifications are of course functional. Also, they are hierarchical. The abstract processor design has approximately a dozen levels and hundreds of individual functions. And just as in a resource, a function by itself may not have meaning, the meaning of a complex resource cannot be separated from its components. Another significant feature of these designs is that they are equally applicable to software and hardware. In fact, the abstact processor design has been implemented in software (Ref. 10). We can now clarify how it is possible to establish the context of meaning required to understand information about a particular part of a design, without requiring a standard means of expression.

In the designs that use the methodology just described , the meaning of any part of the design is determined precisely by two factors. Firstly, there is the general semantics that is part of the context of the specification methodology itself. Secondly, there is the context of elements in the specification that are specifically related to the element of interest. One is the universal context of meaning that is to be applied. The other is the private context of meaning that is relevant to a particular item in the context of its design. In the abstract processor design referenced above, the context of meaning for a specific machine instruction or feature of the processor, can be rigorously determined. With this type of approach to the description of design elements, the contexts that are necessary to establish the meaning of any information in the design can be readily determined. Yet, there is no standard representation of any particular element in the design. (For example, the Boolean resource may have been represented differently). There is also no standard way that elements must be combined. (Different elements may be combined in many ways to achieve the same functional result). Yet, we have indicated that there can be a standard methodology of description and semantics that allows us to determine the meaning of something no matter how it is expressed. It is this kind of approach to the problems of design system integration and interfacing that we are suggesting.

Another way of expressing the same result, is that there need to be standardized methodologies crossing system design boundaries that provide a capability for systematically understanding design information from one system inside another, even though the two systems may support different views, and different technologies of design.

5. Concluding Remarks.

We have raised a number of points that relate to the problems of interfacing or integrating a variety of design systems. The semantic issue includes the problems that arise when there is a gap between what a component is in a system compared to what it is in the mind of a user of the system. Interface difficulties will remain as long as serious semantic gaps remain. We have suggested that the meaning of information in a design system can be represented to some extent in the system, but always includes elements in a broader context of meaning. We have also suggested that universal standards of description for design information seem premature, although some methodological standards may be possible. In particular, a standardized semantic methodology for design descriptions that crosses system boundaries seems to be a reasonable alternative. In general, greater attention to issues of semantics and contexts of meaning seem warranted.

For the functional dimension of design, we have indicated the importance of the concept of a resource. We have also suggested that a theoretical basis for a general theory of functional design that can cross application boundaries is within our grasp. Although, such a prospect seems reasonable in the functional dimension, fundamental problems in the physical and temporal dimensions remain. There are as yet no widely accepted notions of the general meaning of concurrency and communication. There is certainly no acceptable semantic context for these notions, although there are several efforts in this direction.

Finally, we suggest that more attention be paid to fundamentals. There is an ever increasing need for broader perspectives on the design process, and movement away from ad-hoc methods.

References

1. Goguen, J. A., Thatcher, J.W., and Wagner, E.G., " An initail algebra approach to the specification, correctness, and implementation of abstract data types.", In Current Trends in Programming Methodology IV, Data Structuring, R. T. Yeh, Ed. Prentice Hall, Englewood Cliffs, N.J., 1978, pp. 80-149

2. Bjorner, D. "Abstract Software Specification, Proc. Winter Copenhagen School, Springer Verlag, Heidelberg, 1979.

3. Ehrich, H.D.., "On the Theory of Specification, Implementation, and Parameterization of Abstract Data Types", JACM , Vol. 29, No1. January, 1982, pp. 206-227.

4. Ehrig, H., Kreowski, H.J., Mahr, B., Padawitz, P. "Compound Algegraic Implementations: An Approach to Stepwise Refinement of Software Systems" Symposium on Mathematical Foundations of Computer science, 9th Rydzyna, Poland, 1980.

5.McLean, John, "A Formal Method for the Abstract Specification of Software" JACM, Vol. 31, No3. July, 1984, pp.600-627

6. Wegner, Peter, "Reflections on Capital Intensive Software Technology". Software Engineering Notes, ACM Sigsoft, Vol. 7, No. 4, October, 1982.

7. Yurchak, John M., "The Formal Specification of an Abstract Machine: Design and Implementation", Master's Thesis, Department of Computer Science, Naval Poatgraduate School, Dec., 1984.

8. Hunter, James E., "The Formal Specification of a Visual Display Device: Design and Implementation", Master's Thesis, Department of Computer Science, Naval Poatgraduate School, June 1985.

9. Zang, Klaus Harald, "The Formal Specification of an Abstract Database: Design and Implementation", Master's Thesis, Department of Computer Science, Naval Poatgraduate School, Dec., 1985.

10. Davis, Daniel, and Yurchak, John, "The Specification, Design and Implementation of an Abstract Processor", Procs of the 19th Asilomar Conf. on Circuits, Systems and Computers, Nov., 1985.

Acknowledgements

This research has been supported in part by the Naval Weapons Center, China Lake, the Naval Ocean Systems Command, San Diego, and the Foundation Research Program at the Naval Postgraduate School.

DISCUSSION

R.DeSipio, US

It is difficult to communicate with words because:

- Words change meaning relative to culture/education.
- Fads come from the top; trends come from the bottom.

Please comment.

Author's Reply

Methodologically, we need to acknowledge the importance of the semantic contexts that give meaning to words and diagrams (i.e., forms of expression). I generally agree that fads come from the top and trends from the bottom.

Successfully shared, precise semantic contexts do exist. This is what mathematics is all about. Can we develop some rigorous semantic context for design (i.e., a design science)? Would anyone advocate that we can do engineering without mathematics?

W.H.McKinlay, UK

What can we learn from the personal qualities and styles of successful engineers, and what are the implications for education?

Author's Reply

The unquantifiable characteristics of successful engineers are something we really cannot seem to address. Can we, on the other hand, address fundamental design principles that apply broadly and quit rediscovering them over and over? We can sense that such principles must exist, but what are they, and how can they be expressed and become general knowledge? There is still no substitute for creative engineering.

A LOOK TOWARD THE FUTURE OF COMPLEX AVIONICS SYSTEMS DEVELOPMENT USING THE USAF TEST PILOT SCHOOL'S AVIONICS SYSTEMS TEST TRAINING AIRCRAFT

by

Major William H. Broome, Jr., USAF
USAF Test Pilot School
Edwards AFB, CA 93523

and

Mr. Mike Parrag
Arvin/Calspan Corporation - Flight Research Department
Buffalo, NY 14225

SUMMARY

Two important issues exist in addressing the solutions to future avionics systems development problems: avionics systems training for both designers and testers, and the avionics systems development process itself. The airborne avionics training and integration laboratory, such as the USAF Test Pilot School's Avionics Systems Test Training Aircraft (ASTTA), may be a potential remedy for some of the underlying problems of avionics systems development.

ASTTA is a special configuration of the NC-131H Total In-flight Simulator (TIFS), and was developed to fill a significant gap in the education and experience of the avionics systems test community. It provides a cost-effective means of quickly exposing both designers and testers to the key issues of systems development and in-flight testing, especially the operator to systems interface human factors issues. Its benign flight environment is conducive to both initial and advanced training in flight test techniques.

ASTTA offers exciting potential as a test bed for a large variety of research and development activities. Proof-of-concept testing is particularly attractive with minimum packaging required of test devices and the ability to carry several observers and even the designers themselves.

INTRODUCTION

The future development of complex avionics systems is in the hands of two groups of people - the designers and the testers! In the past, these groups have generally performed as separate entities; their viewpoints of the ultimate operational user's environment and functional needs have often been inaccurate. Avionics systems have been developed and deployed which were unusable by the operators due to unacceptably high workloads. Avionics systems have been rushed into testing and operational use without adequate functional integration and simulation with operator-in-the-loop evaluation. Though the situation is slowly improving, problems still exist. Two important issues exist in addressing the solutions to these avionics systems development problems: avionics systems training for both designers and testers, and the avionics systems development process itself. These issues are the focus of this paper. They are also the underlying reasons behind the development of the USAF Test Pilot School's Avionics Systems Test Training Aircraft (ASTTA). ASTTA will be used as an example of a tool to potentially remedy some of the problems in avionics systems development.

Both the designers and testers of avionics systems are often limited in the scope and breadth of their training, knowledge, and experience. Avionics systems designers do their work in the serenity of their office or laboratory, with limited attention paid to the practicalities demanded by the operational environment. Testers are equally guilty! The testers are highly proficient in performance and flying qualities flight test techniques, but often lack avionics systems test training. They remember the cliché for avionics systems testing - "you must invent your own flight test techniques." Too often they step into the cockpit unprepared and unaware!

The avionics systems development process itself may also be the source of a problem. Designers develop systems that work well individually and appear to integrate adequately on paper and in the laboratory. The fact that a human operator is part of the operational system is often forgotten during the functional development of "black boxes" and during their integration into a highly flexible system. Too much flexibility can become detrimental when it causes excessive demands on operator dexterity and mental capacity. The designs must accommodate tactical system operators working in a strenuous environment, not computer wizards or video game players. Often ground verification testing and functional ground simulation are quickly followed by full up flight test in the vehicle with little or no ground-based operator-in-the-loop integration testing. The result is a costly fly-fix-fly schedule that progresses at a snail's pace.

The concurrent integration of the ARN-101 avionics update and the PAVE TACK weapons system to the USAF F-4 was an example of the type of integration problems one would like to avoid. The simultaneous integration of these two systems to the F-4 lacked the

required integration to each other. The result was integration flight testing as opposed to developmental flight testing.

The F-16 Multistage Improvement Program (MSIP) with Block 25 avionics is another example of a process that resulted in a system that was not tailored to the flight environment and fostered operator errors. As an example, it was common for the pilot to have changed his radio frequency inadvertently two to three times before takeoff. Consider what could happen in combat if a pilot could not transmit on the radio to his wingman. The development of the Block 25 systems was done in system integration laboratories which tested only individual black boxes. The result was a good idea poorly implemented and unsuitable for a production aircraft. As a result of serious deficiencies uncovered during flight test, it was necessary to remechanize the avionics prior to production (Block 25B).

These two situations were the product of a process that produced an operationally usable system only after considerable developmental problems were solved. Consider the advantage of adding an airborne simulation laboratory to the development process as shown in Figure 1. One may see a practical advantage but question the cost and scheduling aspects. To put these considerations in proper perspective, one must evaluate the tradeoff between cost and schedule problems of using such an airborne testbed versus developmental risk as shown in Figure 2.

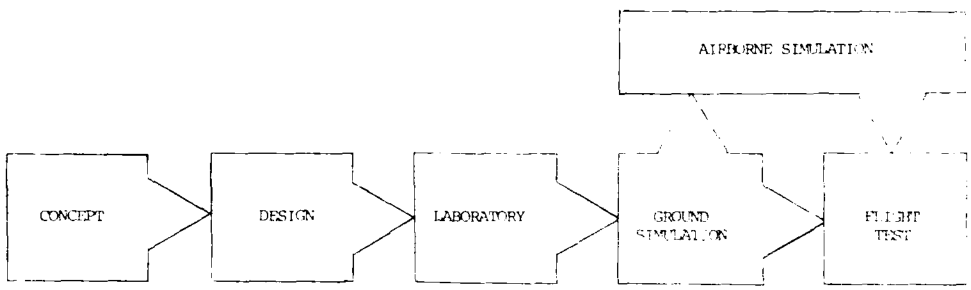


Figure 1. - Avionics Systems Development Process

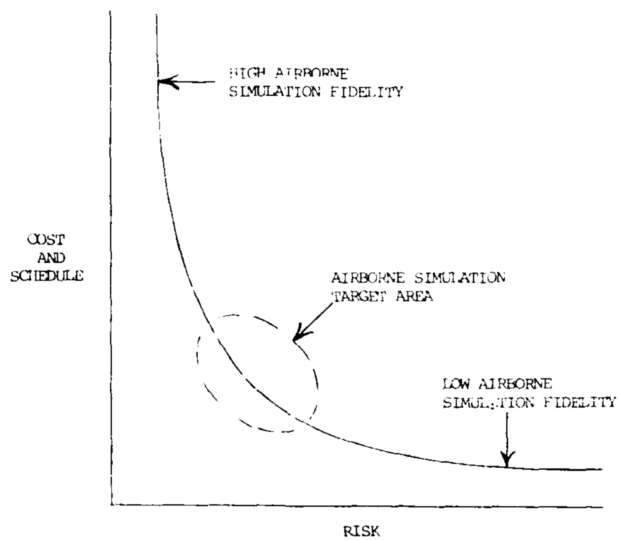


Figure 2. - Airborne Simulation Fidelity Assessment

The fidelity with which such an airborne simulation laboratory represents the operational environment is also a critical concern. High fidelity of environment simulation (approaching 100 percent) may require an exact test vehicle which may not exist when you need it or may be prohibitively costly and/or schedule prohibitive. On the other hand, a lower fidelity simulation (60 to 70 percent for argument's sake) may suffice for proof-of-concept evaluation, and integrate the operator in a relatively realistic environment at an early stage and for significantly lower cost. The question is: What fidelity of simulation is acceptable for developmental assessment? An airborne environment simulation with the knee of the curve (in Figure 2) as its target may serve as the best compromise between risk and cost/schedule, and hence may be the best approach to proof-of-concept testing and airborne simulation.

Ultimately, the avionics systems package must be tested in the vehicle for which the package is being developed. However, many of the new state of the art systems are being developed for new tactical vehicles which themselves are under development. This leads to a dilemma about the timeliness of avionics systems testing in the flight test vehicle. The "build-up technique" is the classical approach to flight test of a new vehicle and its systems. Performance, flying qualities, and clearance of the structural flight envelope initially take precedence over systems testing, and rightly so to some extent. However, cost and time effective development of complex avionics systems requires the interjection of dedicated systems development test and evaluation (DT&E) and initial operational test and evaluation (IOT&E) testing earlier in the flight test sequence than has generally been the case.

Lack of timely operator-in-the-loop simulation and in-flight testing has caused many avionics development scenarios to evolve as follows. First, the testers "curse" the designers for a system that fails to perform adequately in the airborne arena or performs adequately, but with an intolerable workload. Next, the operational users "curse" the testers for recommending changes to the integrated system that either they do not want or which will not work in the operational environment, or which fails to consider the man-machine interface at the lowest common denominator - the new second lieutenant! The designers retaliate with solutions to all these claims, but at the expense of the managers' cost and schedule. The managers respond: Is it safety related? If not, forget it! This vicious cycle continues until initial operational capability (IOC).

Until recently, few solutions have been proposed to alleviate these developmental headaches. About two years ago, the USAF Test Pilot School sponsored the development of the ASTTA training and test facility to help resolve some of the problems we have outlined. The introduction of the ASTTA facility, we submit, represents an important step in the right direction, though not a remedy for all our avionics development problems.

ASTTA was developed primarily to fulfill the training requirements of the USAF Test Pilot School in the avionics systems flight test environment, i.e., train the testers. The evolution of ASTTA concentrated on its objectives of being a familiarization, demonstration, and evaluation tool for avionics systems development for our Test Pilot School pilots, flight test engineers, and flight test navigators.

ASTTA CAPABILITIES

ASTTA is a new configuration of the USAF Flight Dynamics Laboratory's NC-131H Total In-flight Simulator (TIFS) six degree of freedom variable stability aircraft which has been in research and development (R&D) operation since 1971. Figure 3 shows the two different configurations of the NC-131H aircraft. Calspan Corporation of Buffalo, New York, developed ASTTA under contract with the USAF. The ASTTA configuration has the following capabilities and limitations: test airspeeds between 160 and 250 knots indicated airspeed (KIAS), altitudes between 500 feet above ground level (AGL) and 10,000 feet mean sea level (MSL) except for landing evaluations, normal acceleration from 0 to 2.5 gs, sustained turn rates of 12 degrees per second, fuel for a nominal 2 hour mission, and full instrument flight rule (IFR) capability.

ASTTA can carry five systems evaluators in addition to the two Calspan safety pilots and systems engineer. The systems evaluation crew station, which is just aft of the propeller line of the aircraft as shown in Figure 4, has dual seats with standard IFR flight instruments in addition to radar, infrared (IR), and inertial navigation system (INS) controls and displays. An instructor jump seat has been provided between and aft of the crew station seats along with two observer seats further aft in the cabin.

The avionics systems on ASTTA include the Westinghouse APG-66 digital, multimode, pulse doppler radar from the F-16A/B (Figure 5); the Texas Instruments AN/AAS-36 infrared detecting set (IRDS) from the P-3 aircraft (Figure 6); the commercially available Litton LTN-72R INS (Figure 7); a slewable platform with a color camera which is mechanically and functionally interchangeable with the IRDS (Figure 8); and an interface computer with associated software allowing the integrated operation of these sensors. The integrated operation of the ASTTA sensors include the ability to slave either the radar or IRDS to the other in the radar air-to-air mode or the ability to slave the IRDS to the radar in the radar air-to-ground mode. The independent operation of each sensor allows concurrent side-by-side training and testing of each system.

The ASTTA multiplex (MUX) bus structure is shown in Figure 9. The Interface Control Unit (ICU) acts as the MIL-STD 1553 data bus controller for the integrated system (Figure 10). The radar digibus is the primary radar controller. At the present time, three ports on the bus are not utilized and thus could be used for the addition of new systems.

Figure 3a. - TIFS Configuration

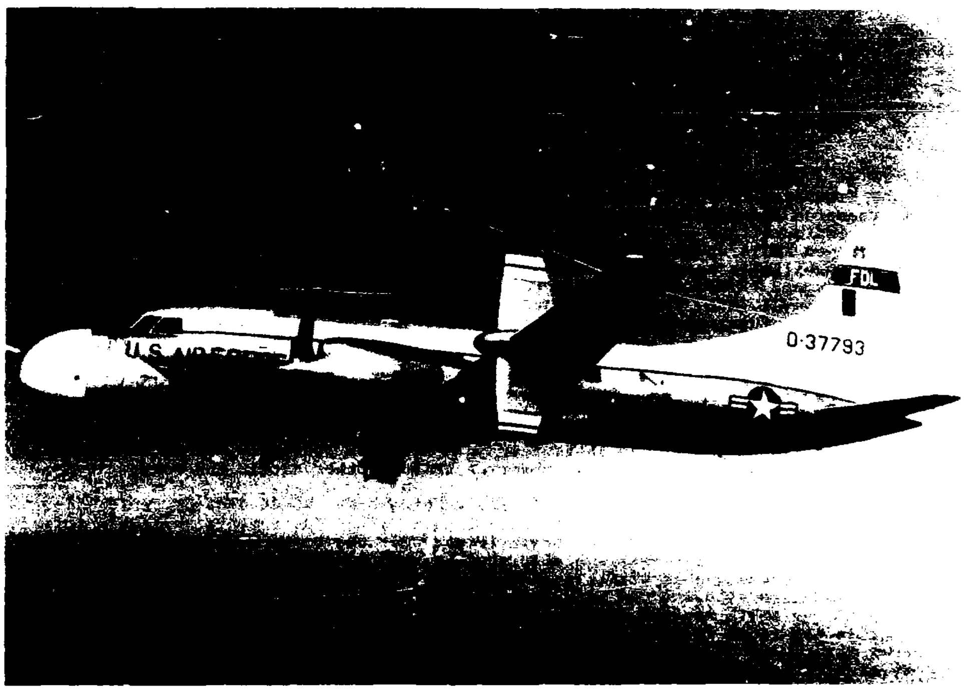


Figure 3b. - ASTTA Configuration

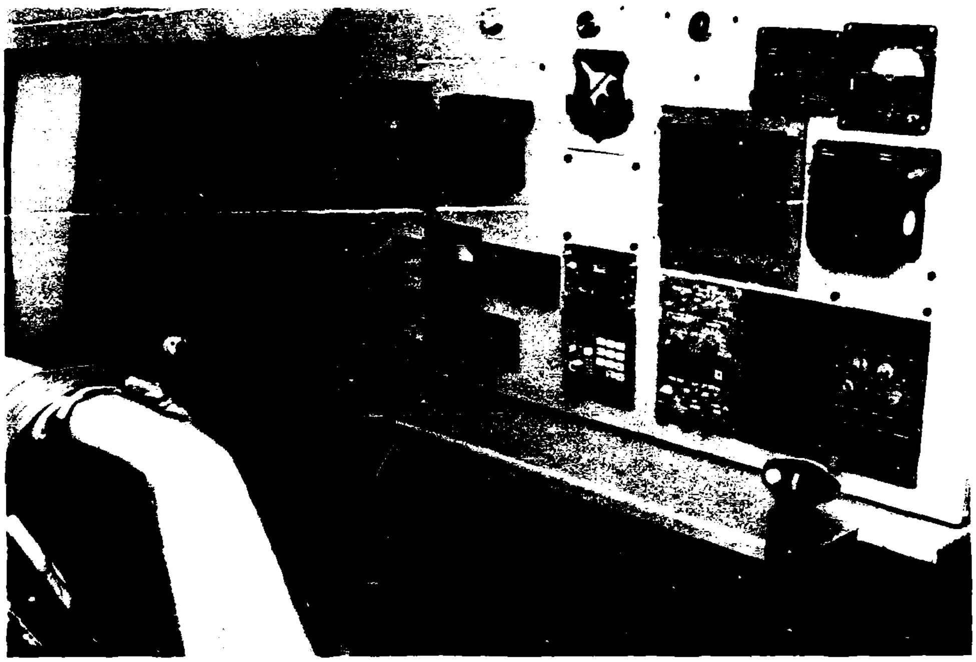


Figure 4. - ASTTA Systems Evaluation Crew Station

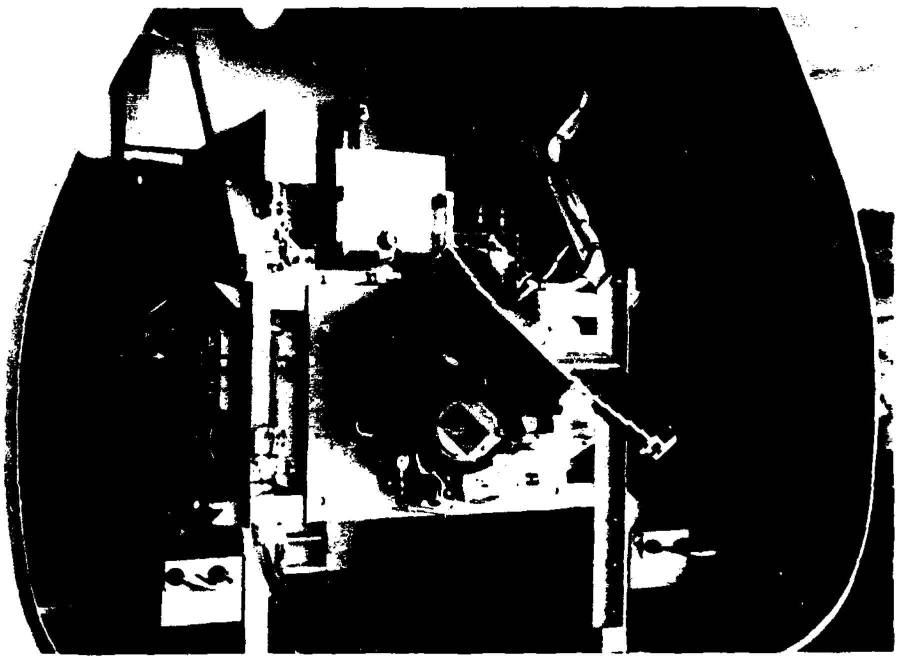


Figure 5. - APG-66 Radar

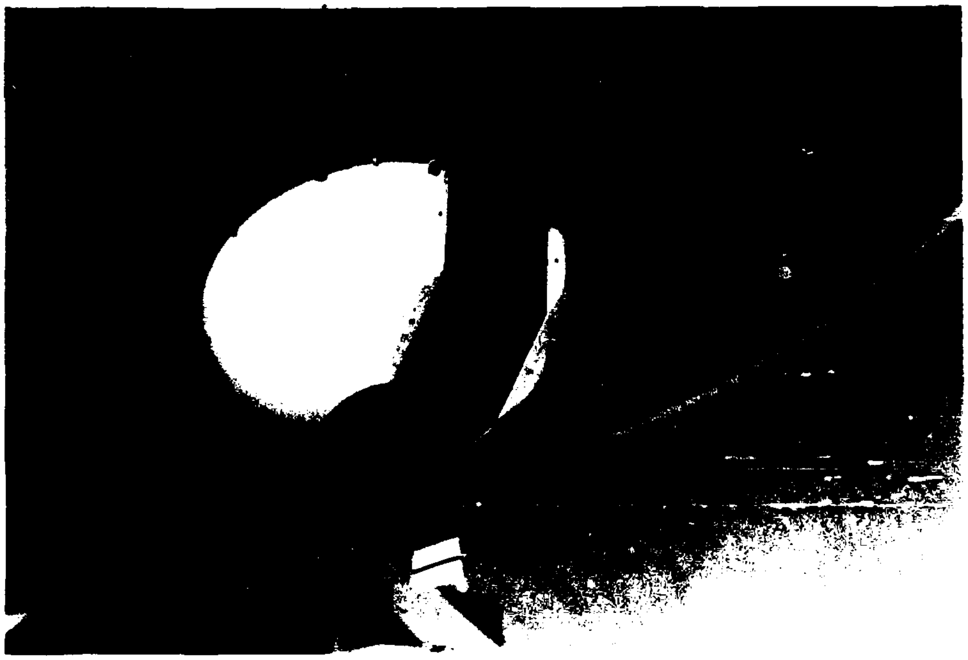


Figure 6. - AN/AAS-36 Infrared Detecting Set

Figure 7. - LTN-72R Inertial Navigation Set Controls and Display

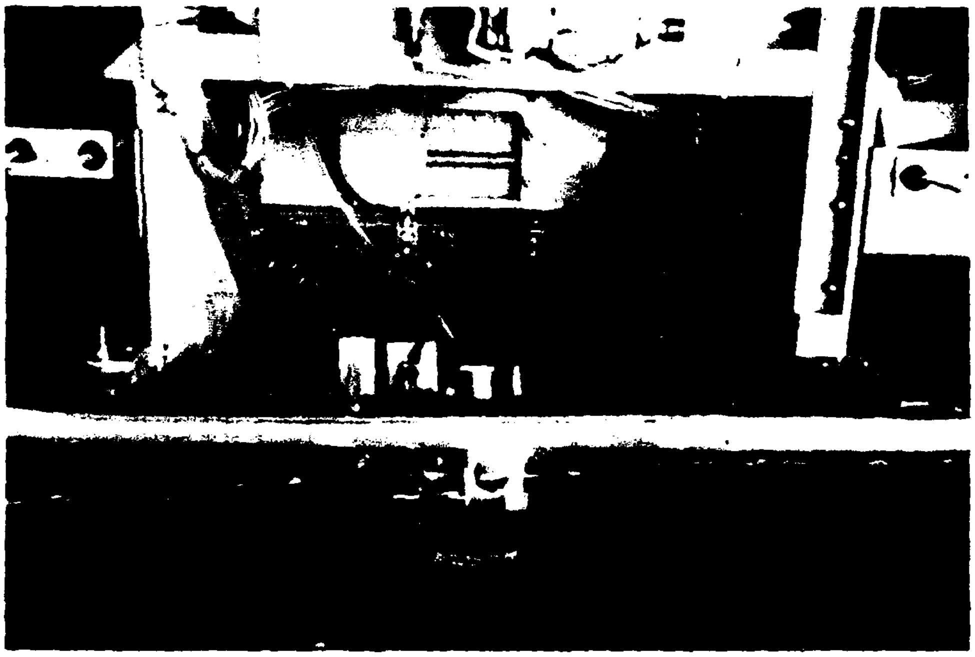


Figure 8. - Slewable Color Camera Platform

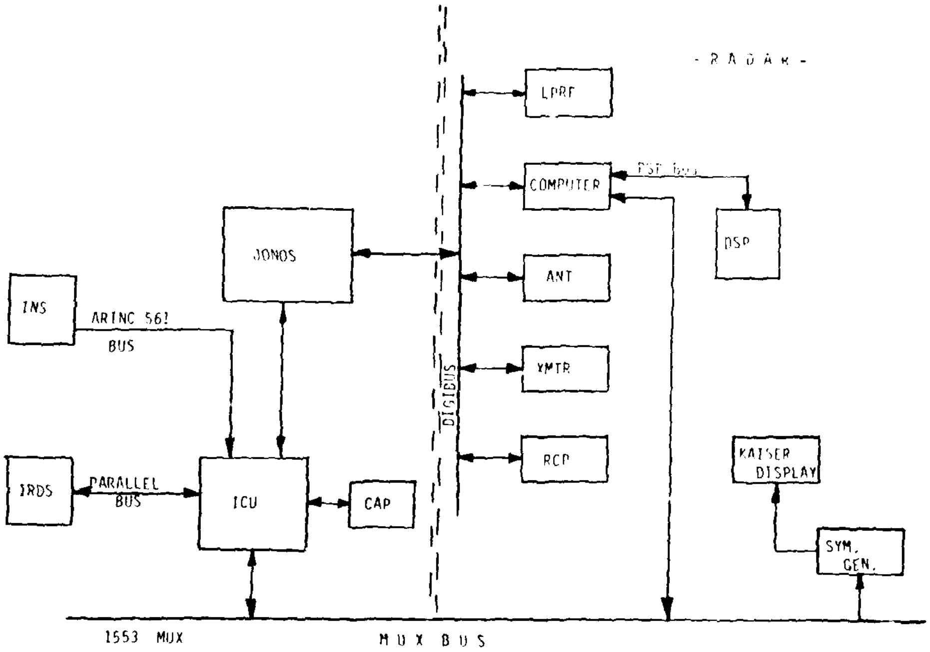


Figure 9. - ASTTA Multiplex Bus Structure

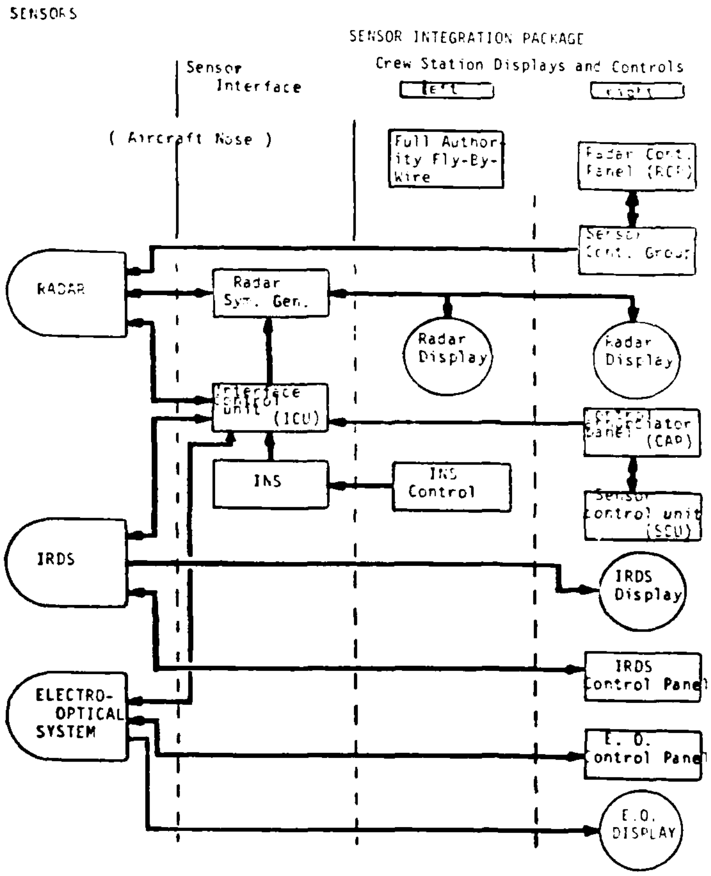


Figure 10. - Functional Diagram of the Integrated Sensor System

The systems engineer station (Figure 11) is the primary station for monitoring and control of the aircraft electrical and cooling as well as all data recording systems. The data recording capabilities of ASTTA include: Video Home System (VHS) recorders for the radar display (525 lines) and high resolution IRDS display (875 lines), for an over-the-shoulder camera pointed at the ASTTA crew station, and for a forward-looking video camera in the safety pilot cockpit for situational awareness at the ASTTA crewstation; a Jonos microcomputer connected to either the ICU or the radar digibus with the capability to read out any 8 parameters for display on a strip chart recorder or for storage on a 58 channel digital recorder; a complete set of aircraft state parameters from the variable stability system (Figure 12) instrumentation (also on the 58 channel digital recorder); voice recording on all VHS recorders; ground radar tracking and positioning using a C-band beacon transponder; and time code correlation of all data through an IRIG B format time code generator, video insertion unit, and cockpit display. Telemetry is planned for the near future. The ASTTA data recording schematic is shown in Figure 13.

With a "fly-by-wire" side stick controller and the capability to fly from either seat (Figure 14), ASTTA offers aircrews a realistic operator workload in a "benign" airborne environment. This "benign" environment includes "time expansion" and a low g environment which are actually advantageous for training purposes and for initial DT&E. When required, sufficient closure rates can be generated with the proper choice of high speed targets. The ASTTA aircraft can sustain turn rates comparable to 400 KIAS/5g conditions. The variable stability system can be used to generate direct lift, direct side translation, turbulence, and other environmental conditions.

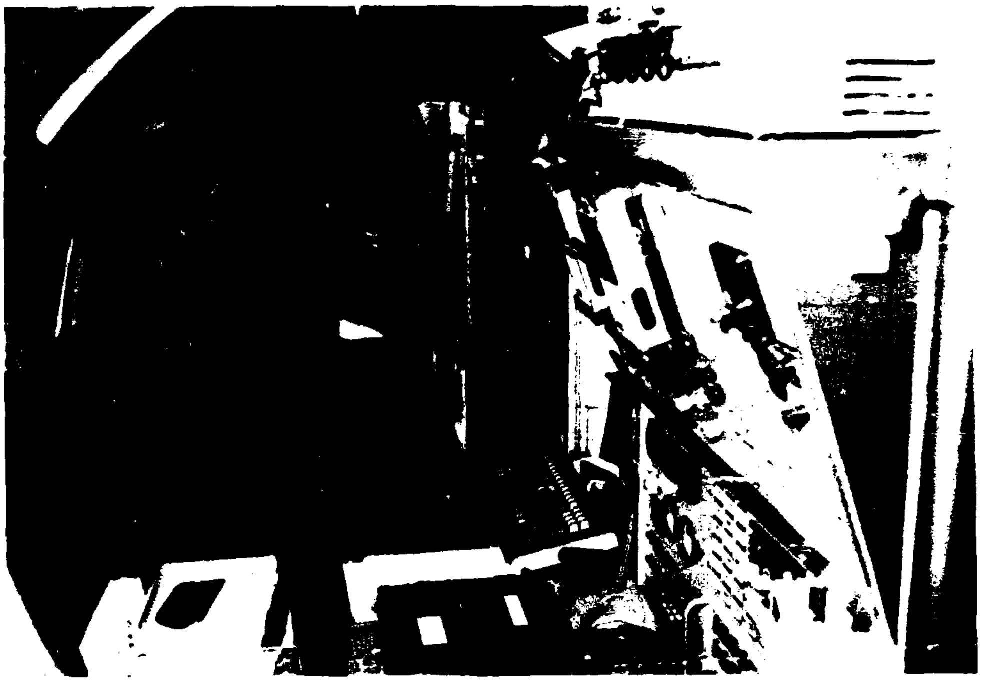


Figure 11. - Systems Engineer Station (Foreground - Looking Aft)

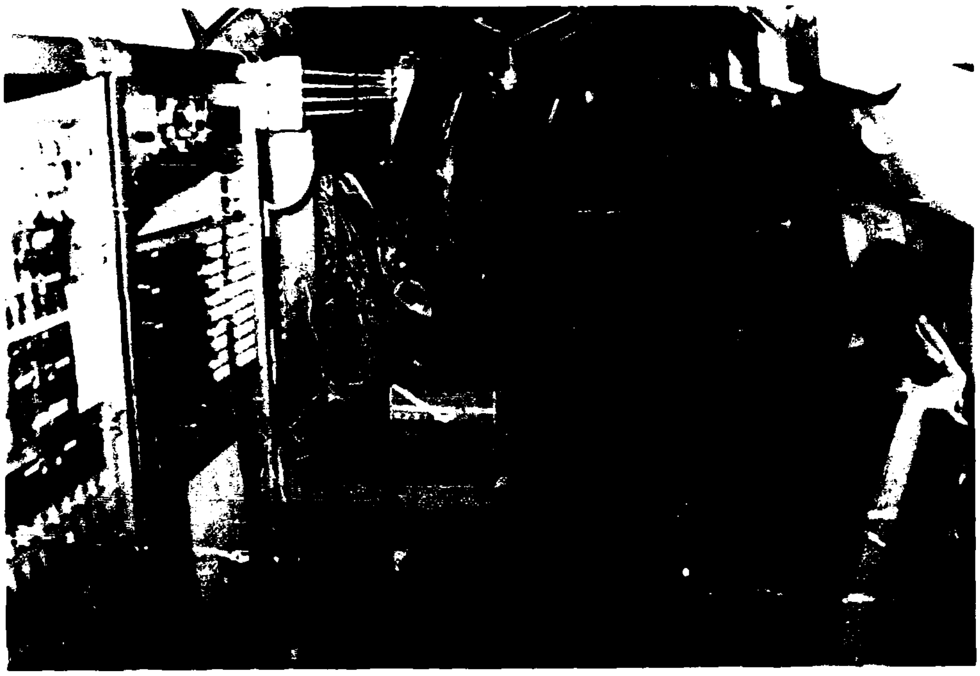


Figure 12. - Variable Stability System Computers (Looking Forward)

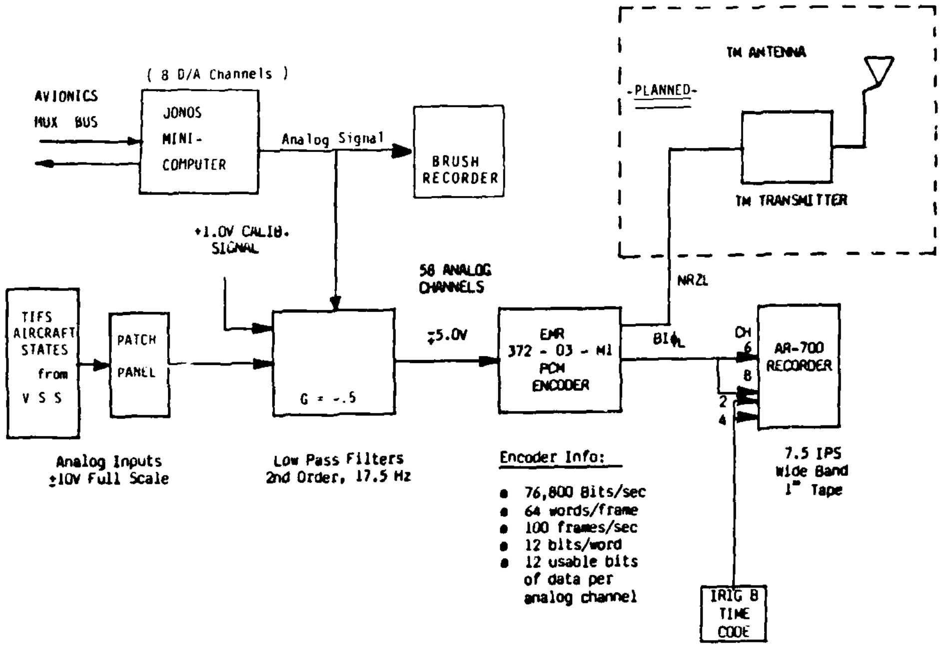


Figure 13. - ASTTA Data Recording Schematic

Figure 14. - ASTTA Fly-By-Wire Side Stick Controller and Seating Arrangement

USAF ASTTA TRAINING CURRICULUM

The USAF Test Pilot School's approach to avionics systems test training is systematic and structured yet flexible to the individual student's prior education and experience. The ASTTA training program is in three phases: ground training, in-flight test training, and reporting, as detailed in Table 1.

Ground training includes: classroom academics on avionics systems and their integration, human factors, and ASTTA operating procedures; laboratories dedicated to demonstrating flight test techniques; and discussion of flight test techniques describing the in-flight maneuvers and data gathering procedures. This is followed by ground familiarization time with the ASTTA systems prior to flight. One major advantage designed into ASTTA is that many of the flight test techniques can be demonstrated on the ground with all systems powered up using ground and airborne targets of opportunity. A weight-on-wheels override switch allows operation of all ASTTA systems on the ground with only the use of external power. Experience has shown that ground run time is limited by the endurance of the operators and not the systems.

Ground Training
(Hours)

	Academics	Laboratory	Flight Test Techniques	Ground Familiarization
Human Factors	6.0	-	-	-
Electro-Optics	8.0	1.5	2.0	-
Radar	11.0	1.5	1.0	-
ECM/ECCM	3.0	-	-	-
INS	9.0	-	1.0	-
Avionics Integration	7.0	-	-	-
ASTTA Procedures	2.0	-	1.0	1.0
Total Hours	46.0	3.0	5.0	1.0

In-flight Test Training

		Hours
Flight #1 (day)	- Air-to-air radar, air-to-ground radar, IR, INS, avionics integration, human factors, and operational navigation route	2.0
Flight #2 (night)	- Air-to-ground radar, IR, INS, avionics integration, and human factors	2.0
Flight #3	- Optional (same profiles as flight #2 or #3)	2.0

Reporting

	Hours
Daily Flight Report	as required
Group Oral Report	1.0

Table 1. - USAF Test Pilot School ASTTA Training Program

In-flight testing includes preflight and postflight ground tests to verify contractor laboratory data; in-flight tests of radar, IRDS, INS, and sensor integration; and the evaluation of pertinent human factors. Table 2 presents a partial list of in-flight test training areas performed during student training.

In the reporting phase, students are taught to accurately report what they saw and to make recommendations to fix problem areas. Mission suitability is always the primary objective. Daily flight reports summarize each mission. The final oral report includes a summary of findings and recommendations from all student training missions. A question and answer period on the oral report and systems test knowledge completes the ASTTA phase of training.

The avionics systems training process is not complete with ASTTA, but is only begun. Students perform numerous other systems evaluations on aircraft such as the A-6, A-7D, A-7K, E-2C, F-4 ARN-101, F-4G, RF-4C, F-14, F-15, F-16, F-18, F-111, P-3, S-3, and T-43.

System	In-flight Tests
Air-to-air radar	- Maximum detection range; minimum and maximum lock-on range at varying aspects and look-up or look-down; time to stable track: elevation, range, and azimuth resolution; track-through-the-notch evaluation; Air Combat Mode evaluation; and mission suitability
Air-to-ground radar	- Minimum and maximum detection range, range and azimuth resolution, ranging accuracy, display accuracy, Doppler Beam Sharpening evaluation, radar low level navigation, and mission suitability
Infrared Detector Set	- Viewing distance; static and dynamic resolution; slew rates and limits; tactical target detection, recognition, and identification; IR low level navigation; and mission suitability
Inertial Navigation Set	- Position error, circular error probable, INS low level navigation, and mission suitability
Human Factors	- Control and display mechanization and compatibility, illumination, atmospheric conditions, noise, procedural sequence, crew station accommodations, ingress/egress, and mission suitability
Avionics Integration	- Radar, IRDS, and INS air-to-air and air-to-ground integration, and mission suitability

Table 2. - In-flight Test Training Areas

USE OF ASTTA FOR SPECIAL PROJECTS

A logical adjunct to the process of training students in flight test techniques on the ASTTA aircraft was the initiation of an advanced systems evaluation project. The first such test project took place in the fall of 1986 with a dedicated systems engineer group. The in-flight task included independent and integrated operation of the sensors while simultaneously flying the ASTTA aircraft via its fly-by-wire system during air-to-air intercepts, and day and night low level navigation routes. The pilot was fitted with physiological instrumentation to yield electroencephalograms (EEG) as shown in Figure 15. The objective of this project was to correlate the pilot's subjective assessment of workload in the in-flight tasks using avionics test modified Cooper-Harper ratings, Subjective Workload Analysis Technique (SWAT), and pilot comments; to the objective measures of workload including EEG, analysis of pilot control activity, and of deviations from the specified altitude and heading profiles. This effort was in support of continuing research in developing objective metrics of pilot workload for use in optimizing pilot-system interfaces.

Figure 15. - Pilot Wired with Physiological Instrumentation (Left - Preflight, Right - In-flight)

It is in the interest of the USAF Test Pilot School to continue and expand this kind of utilization of the ASTTA platform for special projects conducted by the students as practical application of their systems training. Besides giving the students valuable additional hands-on test experience, such projects serve to develop systems test management skills. There is a varied scenario of R&D activities (see ASTTA as an Airborne Systems Integration Test Bed) being contemplated as an evolution of the current ASTTA system, which will readily serve as the basis for future student projects.

MULTI-USER FACILITY

The ASTTA aircraft is a USAF asset available for training and R&D utilization by qualified agencies or organizations. Utilization of ASTTA has evolved, in the short time since its conception, from its original objective of training USAF Test Pilot School personnel to familiarization and training of other Department of Defense (DOD) and contractor testers and designers. The US Naval Test Pilot School is using the aircraft for avionics systems test training of their test pilots, engineers and Naval Flight Officers (NFOs); the Air Force Flight Test Center is using ASTTA for initial and continuation training for its avionics engineers; and Air Force personnel from other bases such as Wright-Patterson AFB, Ohio and Palmdale Air Force Plant 42, California, have trained on ASTTA. One such training session was dedicated to optical system test techniques and was instrumental in the testing of a new optical system for the USAF. Contractor engineers have flown on the aircraft on a noninterference basis as observers. Future dedicated contractor training flights are contemplated with the concurrence and support of their Air Force sponsors

ASTTA AS AN AIRBORNE SYSTEMS INTEGRATION TEST BED

Due to its size, modular design, and extensive instrumentation, ASTTA offers unique opportunities for low cost in-flight test and evaluation of new avionics systems, and in particular, of the interface between the systems and operators (i.e., the human factors issues). Large pieces of equipment and large systems may be tested in a roomy environment with designers looking over the shoulders of the testers or designers doing the testing themselves. "Tweaking" of avionics equipment while airborne is a big advantage of this aircraft. Candidate systems for incorporation in the near term include a Global Positioning System (GPS) to supplement the INS for position information with the accuracy required for new navigation systems evaluations and calibrations, and a Ground Proximity Warning System (GPWS). High on the list of priorities for interface testing are head-up display (HUD) and head down multifunction display (MFD) formats mixing flight status and tactical sensors information in a manner which optimize the displays for specific piloting tasks. Flat plate dot matrix display evaluations are being planned for the near future. A modular "design-a-cockpit" concept with rapid configuration changes will allow cockpit design studies, along with voice activated control studies, and unusual attitude studies using conventional attitude information and new approaches to the spatial disorientation problem such as the Malcolm horizon. Furthermore, with color display hardware integrated, "pathway-through-the-sky" can be presented to the pilot giving threat avoidance guidance which he would "fly" via the fly-by-wire controls. Effective presentation of information continues to be a dominant issue in operator-in-the-loop system development.

Testing of system manipulators/controllers can be performed in a realistic in-flight scenario with an appropriate turbulence environment, either natural or generated through ASTTA's variable stability system. These tests can be performed in a fully instrumented environment, with the pilot/operator instrumented with extensive physiological sensors; while data recorders monitor the avionics systems, the aircraft flight parameters, and the pilot's control activity. This should also provide a valuable facility to test various proposed workload reducing features of the pilot-system interface in highly demanding missions using the single seat night attack fighter scenario as the reference critical bench mark. For example, proof-of-concept testing of pilot associate or expert system devices could be performed cost effectively in a realistic environment.

Although ASTTA's Jonos minicomputer port on the radar digibus is currently passive (data retrieval only), the software can be modified to permit on-line variation of radar operating characteristics for training and R&D purposes. By flying ASTTA over radar test ranges, ECM/ECCM testing can performed. Alternatively, using a data link to ground based computers simulating realistic electromagnetic threat environments, radar performance can be evaluated.

Modification of the ASTTA system bus architecture to include a separate 1553 avionics data bus will facilitate rapid on-line integration of new systems or sensors. This will also permit interaction with the existing basic aircraft flight control system 1553 bus to permit study of flight control/fire control integration issues. The current ASTTA nose with its complement of sensors can be replaced with a mimic nose housing another set of sensors, thereby permitting testing of a variety of sensor combinations and their integration features as shown in Figure 16. The aircraft could also accommodate external sensor pods such as the AGM-65D Maverick, Low Altitude Navigation and Targeting Infrared for Night (LANTIRN), or other new sensor pods, mounted under the nose area.

Other potential test bed applications include: tests of artificial intelligence devices, tactical sensor software development, tests of reprogrammed software of operational systems using the US DOD ADA programming language, optical/IR sensor evaluations on a slewable electro-optical platform, and "blind" landing evaluations using only external sensors for such aircraft as the US National Aerospace Airplane (NASP).

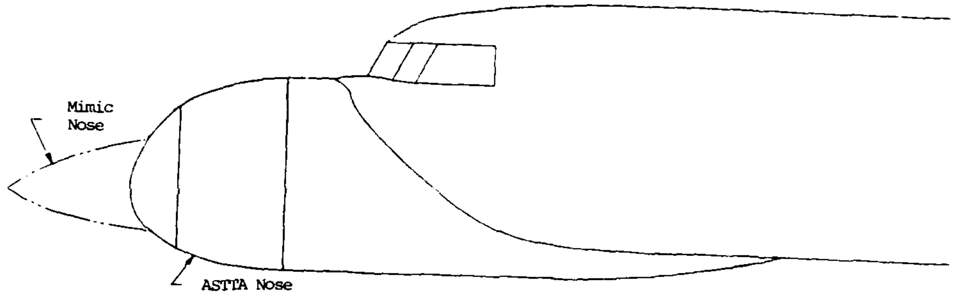


Figure 16. - Mimic Nose on ASTTA (Artist's Conception)

Finally, a plan is being evolved in conjunction with the USAF Human Resources Laboratory (HRL) to incorporate a fighter cockpit with state of the art displays in the cabin of the ASTTA aircraft. Surrounding the cockpit will be a simulator visual dome (Figure 17). By having independent control over the cockpit's "motion base", its visual field, and its displays, tests will be conducted to determine requirements for ground simulators with specified objectives in training. With the ability to project on the visual screen either computer generated imagery (CGI) or real world imagery (via the ASTTA's slewable camera systems), studies dealing with simulator fidelity, avionics integration, and human factors issues can be readily performed.

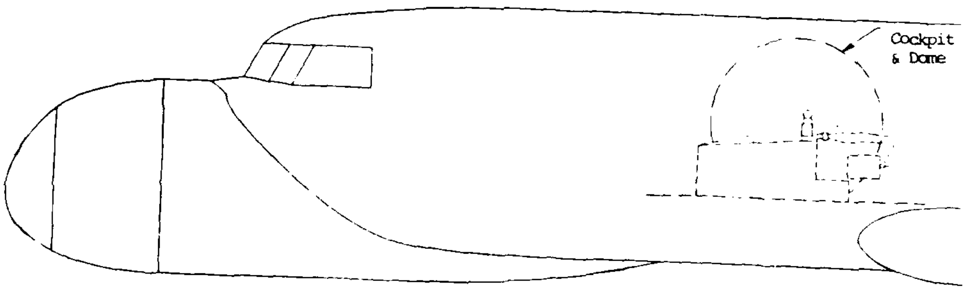


Figure 17. - Fighter Cockpit and Visual Dome Onboard ASTTA (Artist's Conception)

CONCLUSIONS

This paper has examined two important issues in the development of complex avionics systems: avionics systems training for both designers and testers, and the avionics systems development process itself. The tester training issue is being tackled at both the USAF and USN Test Pilot Schools at this time; training for system design engineers, though started to some extent, is in its infancy and may remain there without some emphasis and push from management. Many of today's avionics development problems are the result of attempting to fly without adequate operator-in-the-loop simulation resulting in a fly-fix-fly process which proceeds at a slow and costly pace. In many cases, very limited funding is allocated for training and simulation, yet large sums are spent developing expensive, complex hardware. When simulation is used, often too little time is allotted, or the simulation is inadequate in representing the in-flight environment of the system's intended operation. An airborne avionics training and integration laboratory, such as ASTTA, provides the opportunity to address both the training and developmental issues.

ASTTA is in its infancy as an unique USAF tool to fil_ a significant gap in the education and experience of the avionics systems test community. It provides a cost effective means of quickly exposing both designers and testers to the key issues of avionics systems development and in-flight testing, especially the human factors issues

associated with the operator to systems interface. Its benign flight environment is conducive to both initial and advanced training in flight test techniques. Furthermore, ASTTA offers exciting potential as a test bed for a large variety of research and development activities. Proof-of-concept testing is particularly attractive with minimum packaging required of test devices and the ability to carry several observers and even the designers themselves.

It is the authors' hope that this paper has instilled awareness of the avionics systems training problems and the need for airborne simulation in a realistic operational environment. The future of avionics systems development is being shaped now, and aircraft like ASTTA should be part of that future!

B.Sigaud, FR

Please comment on the problems of using ASTTA for avionics development relative to:

(1) Simulators versus actual hardware.

(2) Time scale expanded.

(3) Nonrepresentation of pilot stress in actual environment.

Author's Reply

(1) Although we currently use actual flight hardware, we could use prototype boxes or even simulate the subsystems of on-board Hawk computers or, if insufficient, use ground computers to simulate subsystem and data link to the ASTTA. (ASTTA provides a real-world signal-to-noise environment for testing.)

(2) Target speeds can be adjusted to reduce time expansion and increase stress. In any case, we advocate the use of ASTTA for early testing. If it does not work in "slow" time, it will not work in a 100-percent real scenario, especially with an operator in the loop.

(3) We advocate complementary use of ASTTA-type tools and tactical aircraft. Also, a very small portion of time is spent at high g levels and high air speeds. The effects of the high g level and high air speed could be extrapolated somewhat from the "benign" environment and then verified in the actual aircraft. However, the "early look" may identify many of the problems you are trying to avoid.

R.J.Young, CA

(1) The requirement for formal systems flight test training is a valid one, and the extension of the principle to designers, as well as to the test community, is interesting and innovative. The use of the platform as a flying test bed, specifically as a simulation facility to support operator-in-the-loop integration testing, is less clear. The capability of systems integration laboratories (SILs) is increasing greatly, and dynamic stimulation is becoming more feasible.

(2) Would it not, therefore, be more effective and less costly to involve the operator more in the original SIL activity rather than to attempt to develop and duplicate the SIL in a test bed and/or associated ground support facility simply as a method to "involve the operator earlier"?

Author's Reply

(1) Yes, the capabilities of SILs are increasing, but the fidelity to properly emulate the airborne environment does not yet exist. Even today's best ground-based simulators are limited in their ability to transfer learning and would have a similar effect in SILs to properly represent the operator's environment.

(2) Yes and no, you must evaluate each development and the risks involved to make a value judgement on the fidelity requirements for airborne simulation. In many cases, it may not require the same effort or duplication of the SIL, since the operator's environment would not have to be artificially duplicated. Much of the SIL hardware/software could be transferred directly to the airborne simulator. Only the system/aircraft interface would differ if you have an airborne simulator with a flexible aircraft bus structure and minimum packaging requirements.

W.E.Howell, US

This comment is in response to Major Young's question. Because of the complexity of software and the time it takes to develop it, it is sometimes cheaper and faster to put a prototype system on an aircraft and fly it. Furthermore, the results are highly credible even to those not involved in the particular technical field being evaluated.

Question:

Are the TIFS aircraft and the ASTTA aircraft one aircraft with interchangeable nose and research hardware?

Author's Reply

Yes, and the strengths of each capability can be used jointly, although the nose change precludes using the fly-by-wire cockpit and the ASTTA tactical sensors at this time.

M.Kayton, US

This is not a question; it is a comment on the avionics design process. In your paper, you say you want to reduce the number of system-level "band-aids" by offering training flights to avionic engineers and managers. Training rides are a fine idea, but in 30 years of avionic system design, I have found that the highest cost band-aids result from premature release of preliminary designs, not from flight control problems. Premature release occurs when the Government program manager or company project manager provides too little money for preliminary design or forces the early release to hardware and software contractors. Often, one more iteration of the design, taking 2 to 4 months, would improve product quality enormously. Instead, premature release leads to delays of a year or more and to costs that are hundreds of times higher than the extra iteration of the preliminary design. Unfortunately, the ASTTA cannot educate the managers on the best time to release the preliminary avionics design.

SOFTWARE ENGINEERING FOR THE BRITISH AEROSPACE EXPERIMENTAL AIRCRAFT PROGRAMME (EAP)

by
W. E. R. Kellaway
Principal Engineer
British Aerospace PLC
Military Aircraft Division
Warton Aerodrome
Preston Lancs PR4 1AX
England

SUMMARY

The software engineering approach adopted by British Aerospace in the specification, design and implementation of the Avionics and Utility Systems Management software for the Experimental Aircraft Programme (EAP) is described.

The software life cycle and supporting methods and tools are described, in particular the Controlled Requirements Expression (CORE) method, supported by the CORE Work Station, and the PERSPECTIVE programming support environment. The considerable benefits obtained in both productivity and quality are highlighted and developments are indicated leading to a full Integrated Project Support Environment (IPSE).

1. INTRODUCTION

In May 1983, British Aerospace (BAe) signed a contract with the UK Ministry of Defence to design and produce a new aircraft as a technology demonstrator, the programme to be known as the Experimental Aircraft Programme (EAP). EAP achieved first flight in August 1986, was demonstrated at Farnborough 1986 and is continuing a programme of flight trials to extend its flight envelope and explore the application of its advanced technology.

The areas of advanced technology are extensive and include an active flight control system, an all-electronic cockpit, and software control of utility systems in the systems area, and extensive use of CFC and super-plastic forming and diffusion bonding techniques in the airframe construction. The programme has involved extensive European collaboration, the advanced technologies having evolved from various government research contracts over the past decade. The technologies will be utilised in any future fighter aircraft programme with UK involvement, the European Fighter Aircraft (EFA) project being a current project.

A less visible aspect of EAP is the successful application of a software engineering approach that had been developed by BAe via a number of small research projects since the late 1970's and for which the EAP project provided the first large scale use. The software engineering approach is the subject of this paper.

2. EAP SYSTEMS ARCHITECTURE

The EAP systems architecture is illustrated in Fig. 1. It consists of 3 major sub-systems:

- Flight Control System (FCS)
- Avionic System (AVS)
- Utilities Systems Management System (USMS)

The AVS and USMS each incorporate a Mil Std 1553 databus architecture and the 3 sub-systems communicate via Remote Terminals on the AVS databus.

The FCS is a full authority digital quadruplex fly-by-wire system. For such a flight safety critical system a significantly different approach is required for the design and development of the software. In particular the use of a low level language and a reduced instruction set is the norm. The EAP FCS is a derivative of the existing Jaguar fly-by-wire demonstrator aircraft system for which a specialised technique had already been developed to support the software production [1]. This technique was used again for EAP FCS and is not discussed further here.

The main feature of the AVS is the all-electronic cockpit with multi-function colour displays generated and moded under software control according to phase of flight and pilot selection. The AVS also includes navigation and communication functions and extensive failure detection and warning handling under software control.

The USMS provides software control of many of the aircraft mechanical systems that have traditionally been controlled by analogue means, in particular the following are controlled via the USMS: undercarriage selection, braking, cabin temperature control and environmental control system, hydraulics, engines, secondary power system, fuel management, fuel gauging and level sensing, and miscellaneous minor systems.

The software engineering approach described in this paper was applied to a limited set of the processors for which BAe had software design lead within the AVS and USMS. Nevertheless this limited set provided a range of 2 distinct CPU types and 9 processor architectures. Within each of the processors software developed by other suppliers was co-resident with the BAe software.

The total software content of the processors considered amounts to some 400K words of which over 300K words were supplied by BAe. However, there exists an amount of duplication within the AVS and USMS so that the total amount of unique software developed amounts to 225K words. For each of the processors BAe was also responsible for integration and acceptance proving of the total software load.

3. EAP SOFTWARE MANAGEMENT

In order to control the software development for a multi-processor, multi-team project, a central software management team was created by BAe independent of the BAe software development team. Thus the BAe software development team was considered in the supplier context and the management team enforced standards and control procedures across all supplier teams. In particular the essential configuration management and quality control roles were performed by the software management team.

Existing BAe standards covering the software life-cycle, already compatible with required defence standards were supplemented with specific standards to take account of the particular methods and tools to be used. Also model texts were produced to specify the documentation requirements for each phase of the life-cycle. The use of project specific standards and model texts ensured a uniform application of the methods and tools across the various teams.

In conjunction with the project standards, the management team also established management control procedures to define the procedures for review, configuration, release for issue and change control mechanisms for documents and software components. Considerable emphasis was placed on achieving quality in the final product both through the "built in" quality inherent through the particular methods and tools used and through the life-cycle review procedures. In addition to the quality control function exercised by the management team to ensure adherence to the standards, a member of the independent Quality Assurance department was seconded to the project. The quality assurance representative reported directly to the project manager on adherence to standards and procedures and carried out an independent check on the configuration standard of life-cycle documents and changes and confirmed that sufficient and necessary testing had been completed before software release.

Additionally, during the software design and code phases an independent technical audit team was employed to review adherence of design to requirements and of code to design, including the effect of changes.

4. EAP SOFTWARE LIFE-CYCLE & METHODOLOGY

The software life-cycle is illustrated in Fig. 2. Although the derivation of system functional requirements may be considered to precede the software life-cycle, the phase is included here because it was supported by the methodology. In essence the approach consists in supporting each phase of the life-cycle by a suitable method with each method or technique in turn supported by a suitable tool. Whilst this philosophy is not new, the scale of application and the degree of integration achieved with the methods and tools on EAP were innovative and contributed significantly to the success of the software development.

The particular methods and tools employed are annotated on Fig. 2 and will be discussed in terms of their life-cycle application in the next section. Two items are singled out here for further discussion as they formed the foundation of the software engineering approach, CORE and PERSPECTIVE.

4.1 CORE

CORE, COntrolled Requirements Expression, arose out of research at BAe aimed at improving the requirements phases of the life-cycle. The research had concentrated on the requirements phase since it was well known that the later that errors are discovered during the life-cycle the more costly they are to fix. Also traditionally requirement errors were not discovered until late in the life-cycle and were considerable in number. Thus any approach that could improve the integrity of the requirement phases would give a considerable leverage on cost reduction. The outcome of the research was the development, in conjunction with Systems Designers plc, of the CORE method.

The CORE method is a top-down approach for analysing and expressing requirements in a controlled and precise manner. It enables a subject requirement to be expressed in terms of lower level requirements which may in turn be subjected to the method to produce a hierarchy of lower levels. The method comprises a number of logical steps which can be summarised as information gathering, proposal of relationships and confirmation of relationships. The steps are described and the method compared with other techniques in [2].

A diagrammatic notation is used extensively in CORE as an aid to understanding and to provide an unambiguous expression of the structure of the subject requirement, the data flows, data dependencies, processing actions and the time ordering of those actions. The notation is illustrated in Fig. 3. Production of CORE diagrams has been automated in the CORE Work Station developed by British Aerospace. The CORE Work Station provides a multi-user development environment for the production of CORE requirements. It consists principally of a diagram editor which allows diagrams to be entered and manipulated via a high resolution graphics terminal, a multi-user database for the storage and retrieval of diagrams and a hard copy facility. Some automatic on-line checking of diagrams is available during editing, and off-line checks are available for more comprehensive checking.

4.2 PERSPECTIVE

PERSPECTIVE is a proprietary product of Systems Designers plc, which provides a multi-user programming support environment for the development of embedded computer systems written in Pascal. The product arose out of considerable experience in the development of cross-compilers and support environments for embedded systems and first appeared on the market place in 1983.

Whilst the maturity of the product was rightly questioned for application in the EAP timescales, for the chosen language Pascal, the features offered by the product were considered to far outweigh any risk associated with immaturity.

PERSPECTIVE supports a particular modular design technique which has been implemented through extensions to the ISO standard Pascal. The technique is based on a data flow model of a real time system, and decomposes the structure of the software into three basic component types, processes, interfaces and modules. The processes are the fundamental units of parallel processing within PERSPECTIVE and consist of independently schedulable units of sequential code. Processes can only communicate with each other via the facilities specified in the interfaces. The interfaces contain only the specification of the facilities available, the implementation of these facilities is defined in the modules. Modules may also use the facilities defined in other interfaces, so that the concept of software layering or successive layers of abstraction is inherent in the technique. The basic components may be grouped into larger components termed subsystems which again can only communicate via interfaces. The technique is supported by a diagrammatic notation which is summarised in Fig. 4. The notation has been automated in the CORE Work Station to provide computer support of the design technique.

PERSPECTIVE Pascal also includes extensions to provide basic primitives for the manipulation of processes to enable the implementation of the required scheduling regime for a particular application. The basic primitives are implemented in a run time system supplied in source code which may be tailored for any particular hardware configuration.

The design concepts and notation of PERSPECTIVE are very akin to the MASCOT approach [3]. Earlier research work at British Aerospace had indicated that a requirement expressed in CORE notation could be mapped into a MASCOT design diagram, so that for EAP it proved a straightforward extension to map the CORE requirements into PERSPECTIVE designs.

Central to the PERSPECTIVE support environment is a multi-user database in which are stored the source code files and all associated derived products for the components of one or more systems. The database is configured as a number of independent user domains permitting independent development of components. Components may be shared between domains through the use of a publish and acquire facility. Components may exist in multiple versions and extensive management facilities are provided to keep track of versions, relationships, status, access rights etc. providing the essentials for configuration control.

Software development within PERSPECTIVE is carried out in two major phases, Host and Target. During the host phase components are compiled and debugged through execution on the host (VAX series) computer. Checkout facilities are built-in to enable source-code level debugging, including single-stepping, breakpoint setting, data alteration and data and system monitoring. For the target phase the components are re-compiled for the target configuration and downloaded via a terminal line to the target processor. Execution in the target may continue independently from the host or the host link may be retained to allow checkout of the target software via the host facilities in a similar manner to host checkout. Complete target memory images may also be generated via a 'PROM' facility within PERSPECTIVE which generates files suitable for subsequent processing by IC programming devices.

5. EAP SOFTWARE TOOLSET LIFE-CYCLE APPLICATION

This section discusses the methods and tools further in the context of their application to the EAP software life-cycle.

5.1 Requirements Definition

Requirements definition encompassed the two phases of System Functional Requirements and Software Requirements. Both phases were carried out by system engineering staff (as distinct from software engineers) utilising the CORE method, implemented in the CORE work station.

Two different approaches were adopted for the AVS and USMS arising from the essentially different nature of the systems. The AVS consists of a highly integrated system where the problem consisted in identifying and elaborating the various functions and then allocating them to defined processing units. The USMS consists of a number of largely independent subsystems which required independent analysis and then integration into a defined processing architecture. The routes taken in each case are shown in Fig. 5. In each case the end result was a set of processor software specifications.

Each processor software specification consists of a set of CORE documentation supplemented by information on hardware/software interfaces, constraints, error detection and handling, scheduling etc. The CORE documentation consists of a set of thread diagrams, which show the required independent threads of execution within the processor, an operational diagram, which shows the required time-ordered relationship between the individual threads, and data decomposition diagrams for complex data items. The diagrams are supplemented by "node notes" which supply brief textual descriptions of each process in a thread diagram and each data line entering or leaving the diagram. The node notes form part of the CORE documentation set and are entered into the CORE database via the work station. A typical thread diagram is presented in Fig. 6.

In addition to the processor software specification, the system engineering staff produced an acceptance test specification for each processor which served as the acceptance requirement for delivery of processor software for subsequent system integration.

5.2 Basic Design

The basic design phase consisted of the translation of each processor software specification into a definition of the modular structure of the software within the processor and an identification of all software components. The basic design was carried out by the BAe software team with participation from suppliers of co-resident software where applicable. The PERSPECTIVE design approach was followed, supported by the CORE work station. In essence the design consisted of a grouping of the CORE threads defined in the software specification into suitable PERSPECTIVE processes and/or subsystems and a grouping of the data items into suitable interfaces with specified access procedures. A typical basic design diagram is presented in Fig. 7.

Also during the basic design phase the required testing plan for the processor was specified. This identified the order of testing of components and the necessary test components to progressively test and integrate the software within the processor leading to acceptance test of the full processor load.

5.3 Detailed Design

The detailed design phase consisted of elaboration of the detail for each software component and test component defined during the basic design phase. For this phase the work was carried out independently by the various software teams, working to the precise specifications derived during the basic design phase.

Since much of the detail was contained in the detailed layering of the processor software specification CORE threads, it was possible to transfer this information into the detailed design supplemented by definition of data types and structures and access procedures. Again the CORE work station was used to transfer the thread information into detailed design diagrams and to add the extra information. Where necessary the diagrams were supplemented by pseudo-code node notes. A typical detailed design diagram is presented in Fig. 8.

5.4 Code, Test, Integration

The code, test and integration phases were performed using the PERSPECTIVE support environment. The PERSPECTIVE database facilities were used to store the code of all design components and test components and the various build standards leading to complete processor software builds. Generally the test and integration phases were as follows:

- Host test of individual Pascal components using test components where necessary and previously tested components when available. For the host tests extensive use was made of the PERSPECTIVE checkout facilities.

- Repeat of the host test on the target processor using the PERSPECTIVE download and checkout facilities.

- Target test of assembler language components using the download and checkout facilities.

- Progressive integration and test of components in the target processor using the download and checkout facilities.

In practice the repeat of the host testing on the target processor was found to be unnecessary as no errors were introduced by recompilation for the target, and this phase was subsequently omitted.

Assembler language components were necessary in some instances to interface with processor hardware or for critical timing areas. Within PERSPECTIVE assembler language is confined to module components.

Two additional tools were employed during the target testing and integration phases. A comprehensive Data Acquisition and Simulation System (DASS) attached to the overall aircraft test rig was used to stimulate and monitor hardware interfaces. Also microprocessor development systems (MDS) were used for processor emulation, state analysis and logic analysis operations. The MDS facilities proved especially useful in resolving hardware/software conflicts.

Whilst software developers could initially carry out informal testing on uncontrolled items for experimental or confidence checking, eventually the required testing was carried out formally on "frozen" components under control of a central software librarian. This was achieved through the PERSPECTIVE configuration control facilities. Each developer was assigned a unique user domain and special protected domains were assigned to the software library. Components ready for configuration control were submitted via a registration procedure to the librarian and transferred to a library domain. Components in the library domain were published and could be acquired by developers for incorporation into test builds as necessary, however the components were effectively frozen and unable to be modified by acquirers. A version numbering convention was enforced by the librarian to accommodate formal changes of components and coding standards included a header comment identifying the version change history of the component. Through these simple but effective facilities, components were progressively transferred from development domains to library domains, building a complete library of components under configuration control with full design traceability.

An important feature implemented as part of the library procedure was an independent technical audit of source code components on registration with the library. This ensured conformity to design documents and formal changes, and conformity to the project coding standards.

As part of the integration task, BAe was required to incorporate the software delivered from other suppliers. The modular structure of the software enforced through the PERSPECTIVE facilities resulted in a virtually trouble free integration of PERSPECTIVE components. More significant problems occurred in certain cases where suppliers had not used the PERSPECTIVE approach, however overall the problem of integrating separately sourced software were considered to be minimal.

Formal releases of software for system rig testing or aircraft use were assembled by the librarian in PERSPECTIVE databases from the component library.

Following acceptance of the releases, the subsequent system rig testing and system integration were carried out by the Systems Test department, a separate organisation independent from the software teams, making use of the DASS facility. For early processor units with RAM storage, the systems test team made use of the download facility to load the units directly from the PERSPECTIVE databases. For later units with ROM storage, memory images were generated by the software library via PERSPECTIVE facilities and subsequently programmed into storage devices.

Errors discovered during system testing were corrected via changes raised on the appropriate requirement and/or design document and affected code components were raised in issue and re-tested. The implications of changes were easy to assess and few side effects were encountered. As confidence was gained in the approach, only incremental re-test was undertaken by the software team on re-release.

6. BENEFITS OF APPROACH

The benefits of the software engineering approach described have been assessed by comparing the results achieved on EAP with the experience at British Aerospace on earlier projects. The main gains are an increase in both productivity and quality. Compared with the two most recent projects, the productivity measured in delivered lines of code per man year for the software design code and test phases has increased five-fold. The quality measured in terms of errors detected during rig flight clearance testing expressed as a ratio per line of delivered code has improved by almost a factor of ten. This increase in quality and productivity has resulted in a considerable estimated saving in the total resources needed for design and rig testing, which outweighs the expense of introducing the methods and tools by a factor of six.

The use of well-defined methods and tools also provided a detailed framework for training, enabling a comprehensive training scheme to be planned which allowed relatively inexperienced staff to be brought up to speed reasonably quickly. The modular approach supported by the methods and tools also allowed the introduction of large numbers of extra staff during the detail design, code and test phases to keep the project on time schedule. In fact the peak software design team manning at British Aerospace reached 60 whereas the maintenance of the software to incorporate requirements changes resulting from flight experience is handled by a team of 4.

7. CONCLUDING REMARKS

The software engineering approach described has undoubtedly contributed to the success of the EAP project in meeting an extremely tight timescale. Also results indicate that the quality of the delivered software and the productivity achieved have both improved considerably through the approach. The firmly held view within British Aerospace is that the use of the CORE method supported by the CORE workstation has been a major factor in the success through ensuring a structured top-down approach resulting in quality, consistent requirements and providing an essential continuity through the software life-cycle into the design phases. PERSPECTIVE represents the state-of-the-art in support environments for embedded applications using the Pascal language and was a further major factor in the success. Both CORE and PERSPECTIVE are in current use on further projects within British Aerospace.

Developments of the approach for future projects are aimed at the introduction of the Ada(R) language and providing a truly integrated project support environment (IPSE). Whilst the methods and tools of the current approach are integrated in the sense that they are compatible and consistent from phase to phase of the life-cycle, nevertheless the products of the requirements, design and code phases are held in distinct databases, each subject to individual configuration management procedures and considerable manual intervention is required to ensure overall configuration control. The IPSE approach aims to utilise a common database implementation tool so that all life-cycle products are subject to a common version and variant control mechanism, thus enabling automated configuration management across the total life-cycle.

Several products are appearing on the market place which provide a basic IPSE framework giving the essential database manipulation and management facilities, and which permit the integration of specific tools via a tool interface.

Currently British Aerospace in collaboration with partner companies is involved in the procurement of a basic IPSE framework and the development and incorporation of specific tools. Two particular areas of development are worthy of note. Firstly CORE has been integrated with EPOS [4] which allows the EPOS facilities to be used for extensive analysis of CORE data and makes available the extensive EPOS documentation facilities. Secondly guidelines are being developed for the translation of requirements expressed in CORE diagrammatic notation into software design structures suitable for Ada implementation.

An IPSE schematic is shown in Fig. 9, however the specific tools indicated are illustrative only. The IPSE development which will form the basis of the software engineering approach for future projects and in particular the European Fighter Aircraft (EFA) will serve to maintain the improvement trends in quality and productivity already observed for EAP.

8. REFERENCES

[1] Daley E., Smith R. B. : 'Flight Clearance of the Jaguar Fly-by-Wire'. Certification of Avionic Systems, The Royal Aeronautical Society, London. April 1982.

[2] Price C. P., Forsyth D. Y. : 'Practical Considerations in the Introduction of Requirements Analysis Techniques'. Proceeding No. 330, AGARD Software for Avionics. Sept. 1982.

[3] 'The Official Handbook of MASCOT' MASCOT 2 Issue 2. Royal Signals and Radar Establishment, Malvern. March 1983.

[4] Biewald J., Goehner P., Lauber R., Schelling H. : 'EPOS - A Specification and Design Technique for Computer Controlled Real-Time Automation Systems'. Proc. 4th Int. Conf. Software Engineering, Munich 1979. IEEE Comp. Soc., Los Alamitos, Cal., 1979, pp 245-250.

9. ACKNOWLEDGEMENT

This work has been carried out with the support of the Procurement Executive, Ministry of Defence.

(R) Ada is a registered trademark of the U.S. Government - Ada Joint Program Office.

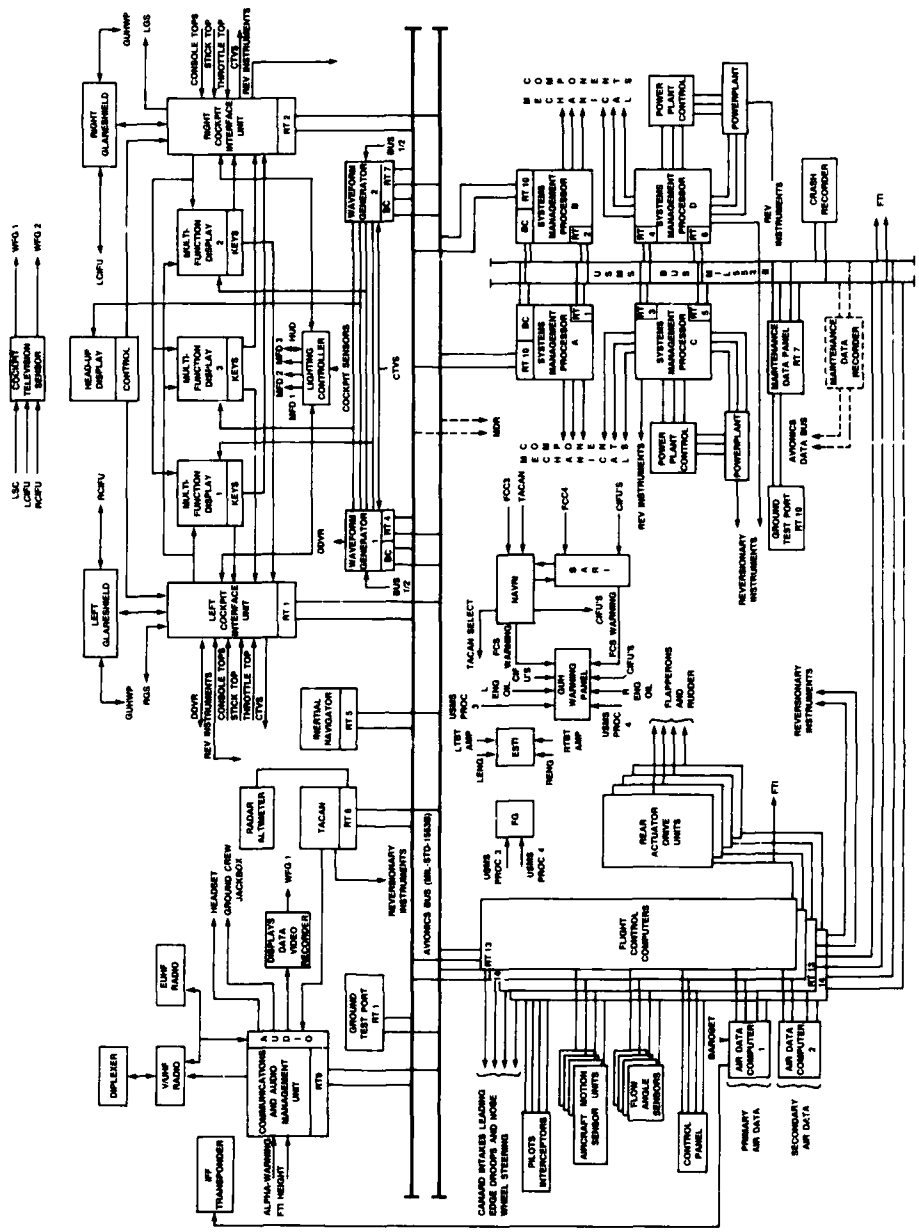


FIG.1 EAP SYSTEMS ARCHITECTURE

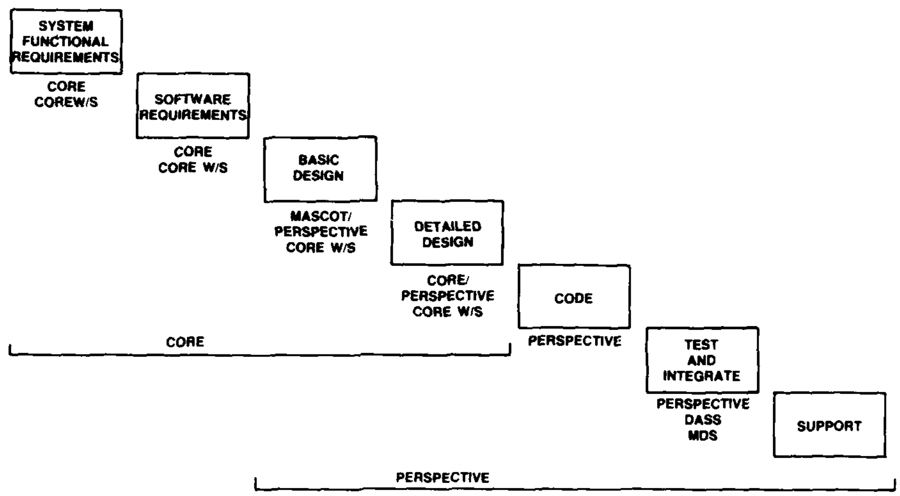


FIG.2 SOFTWARE LIFE CYCLE

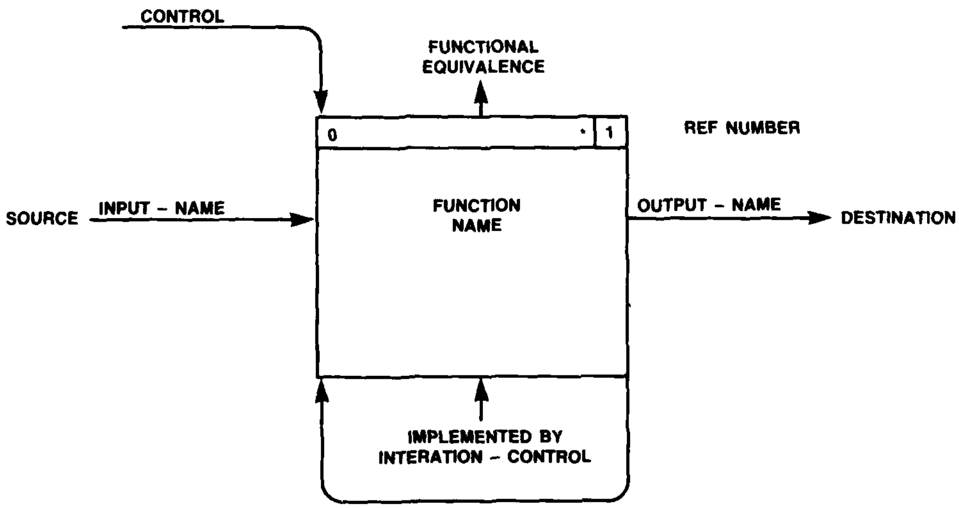


FIG.3 CORE NOTATION

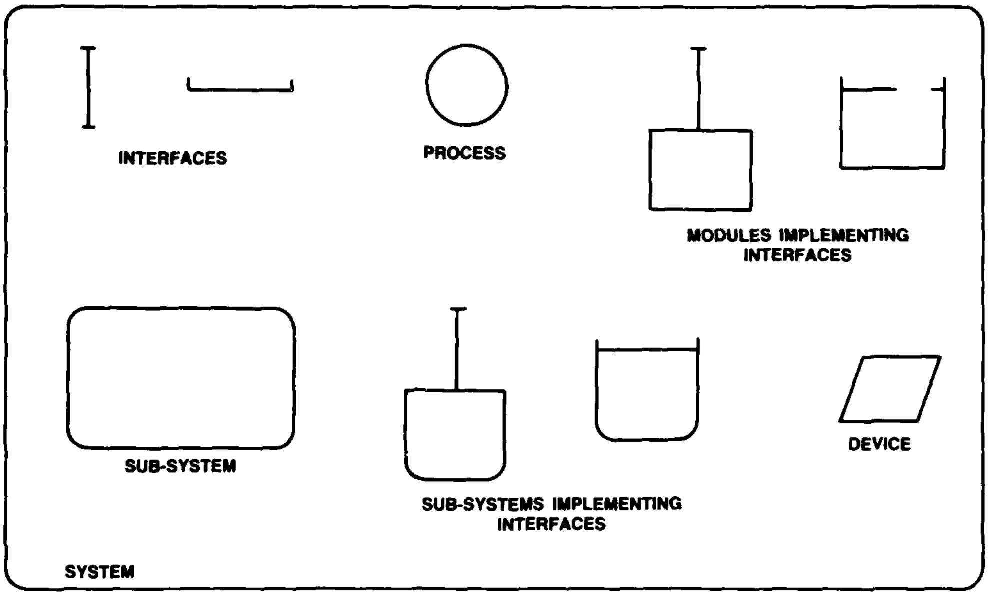


FIG.4 PERSPECTIVE NOTATION

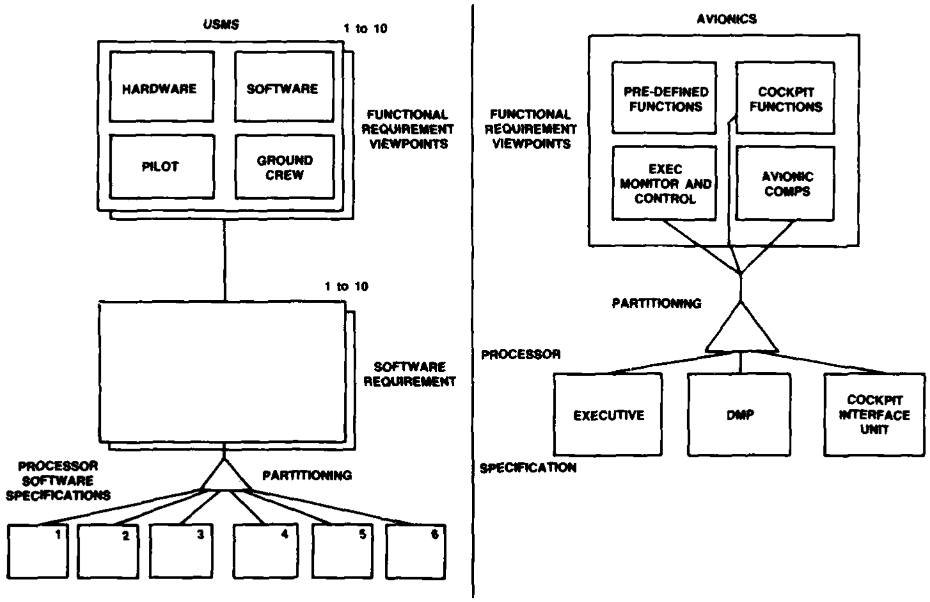


FIG.5 SOFTWARE REQUIREMENT ROUTES

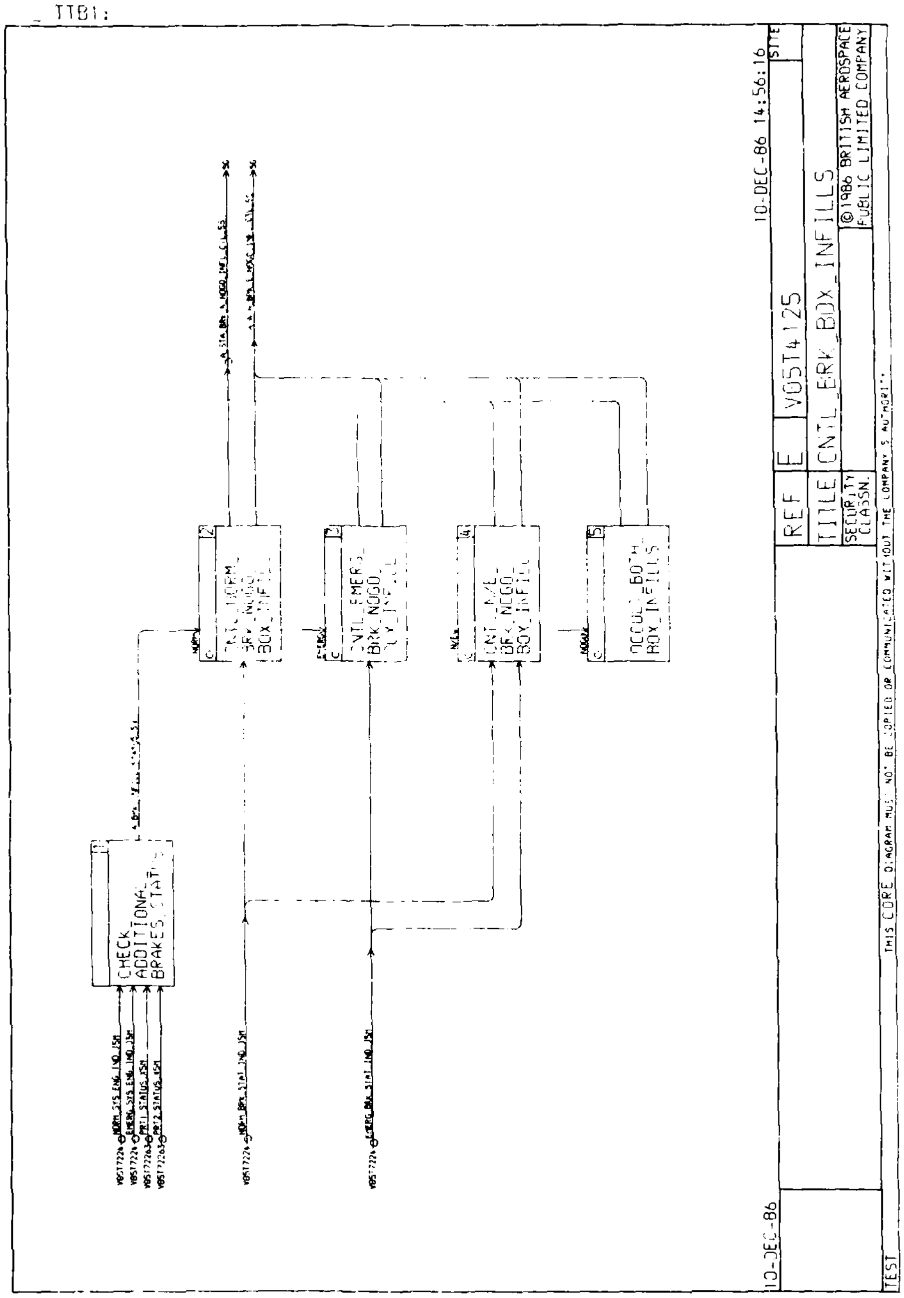


FIG.6 EXAMPLE CORE THREAD DIAGRAM

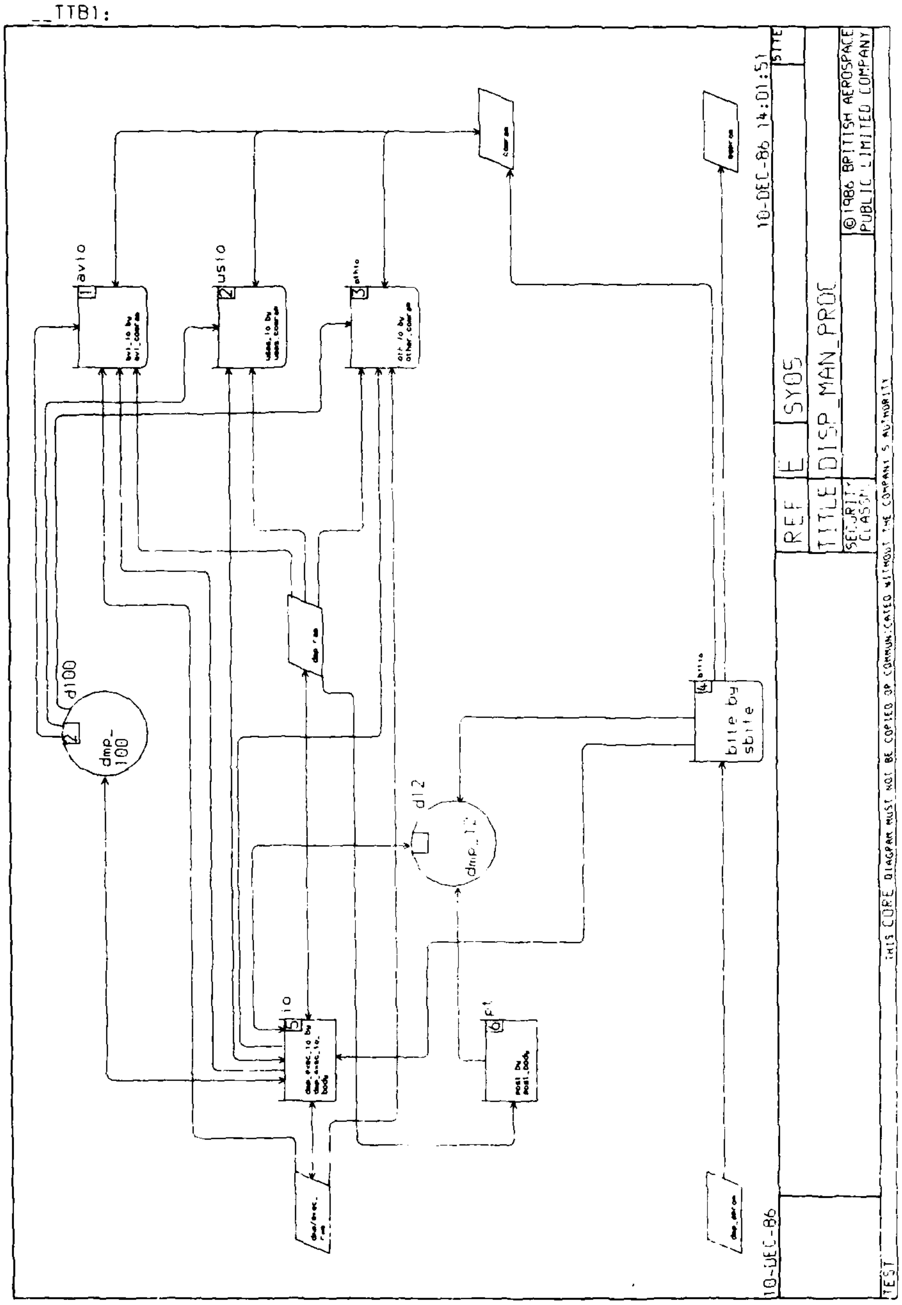


FIG.7 EXAMPLE BASIC DESIGN DIAGRAM

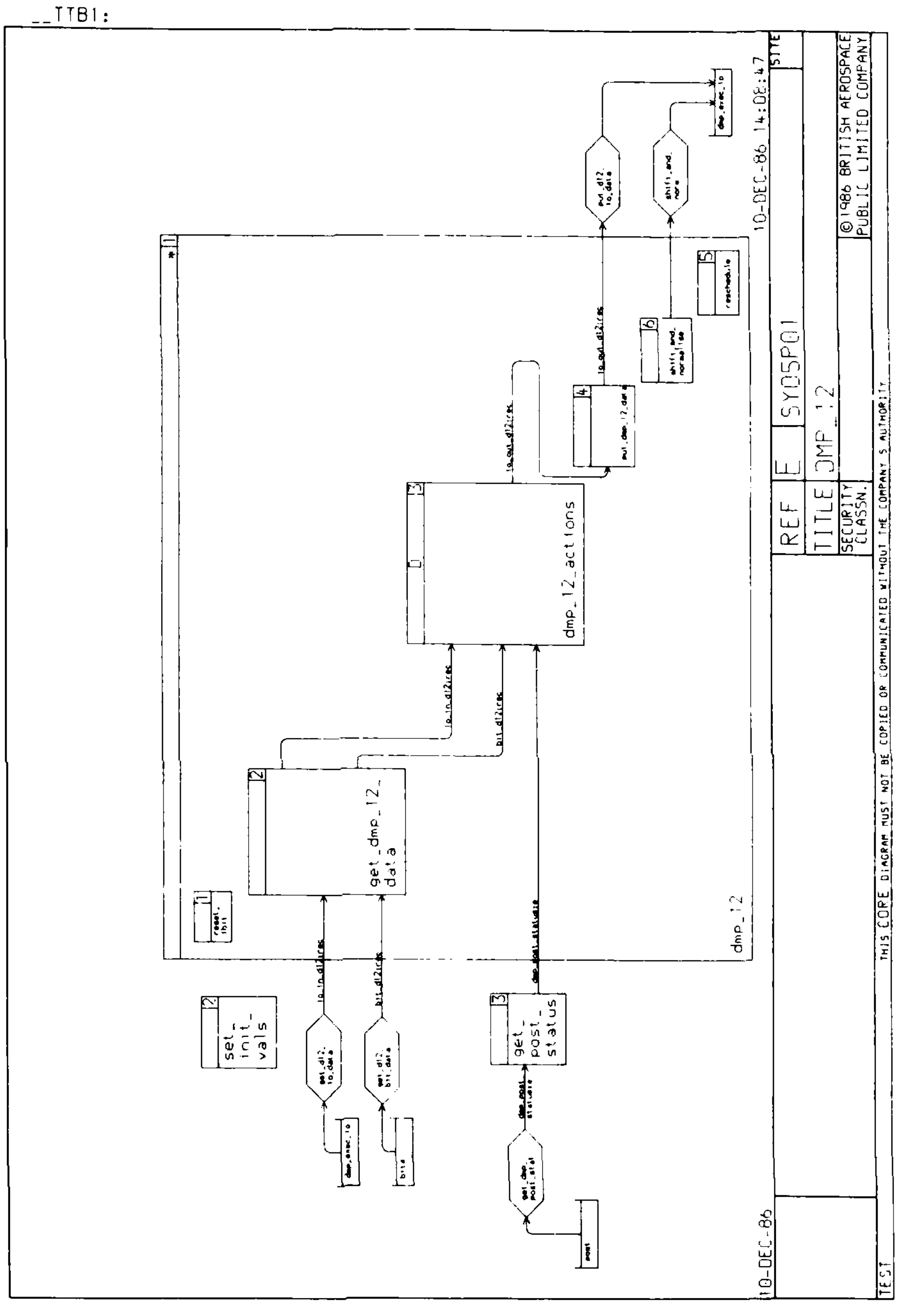


FIG.8 EXAMPLE DETAILED DESIGN DIAGRAM

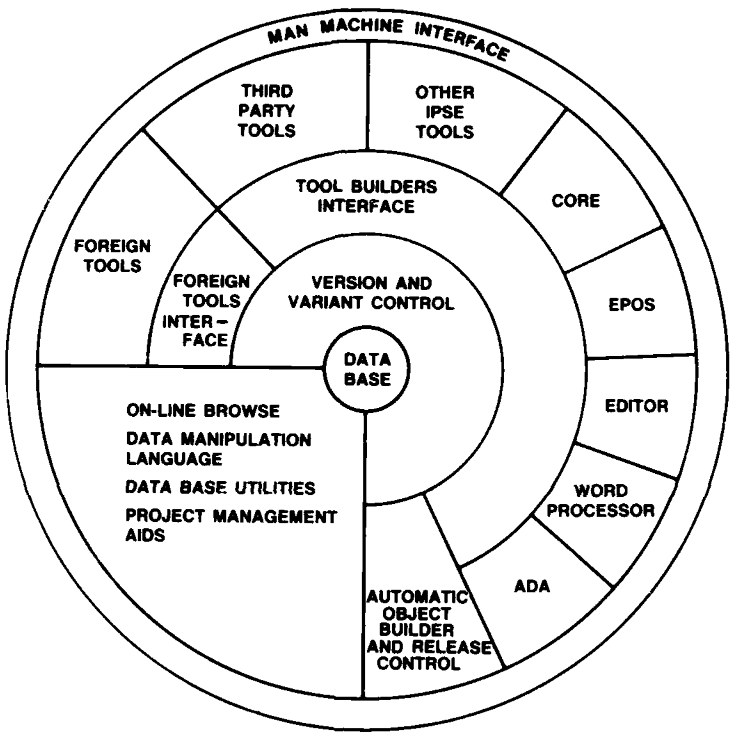


FIG.9 IPSE SCHEMATIC

DISCUSSION

M.Muenier, FR

You indicated that for the EAP, both CORE and MASCOT are used at the design stage together with real-time Pascal as an implementation language. On the other hand, you plan to use EPOS for design in the framework of the EAF program with Ada for coding. Would you comment on the independence of design tools with regard to programming languages?

Author's Reply

For EAP, we choose CORE/MASCOT/PERSPECTIVE; PERSPECTIVE provided the support environment for Pascal.

For EFA, the required language is Ada, and work is under way to produce the Integrated Project Support Environment (IPSE) that will embrace Ada. The system design tool will be CORE/EPOS, where the design advantage of CORE will be coupled with the cross-checking advantages inherent in EPOS.

I believe the design tools should be capable of supporting several programming languages; however, with an aircraft program like EFA, an Ada-dedicated IPSE would be considered acceptable.

P.R.Walwyn, UK

(1) How "modular" was the software you produced? What sort of "tasks" did you include in discrete modules?

(2) What proportion of your total code consisted of "repeat-calls" of common modules?

(3) How "reusable" do you think your produced modules will be for future programs (e.g. EFA)?

Author's Reply

(1) The software is highly modular, in which the module size varies considerably, covering items such as monitors or track compute or display drive logic.

(2) Out of a 300K word software package, 225K words were unique.

(3) The software documentation requirement can be used again, since it is independent of project. However, EFA will be Ada-based, and the software modules produced for EAP in Pascal will not be reusable.

SYSTEME AVIONIQUE – METHODE DE DEVELOPPEMENT ET OUTILS INFORMATIQUES

par

P.Laroche-Levy
Avions Marcel Dassault
78, Quai Marcel Dassault
92260 Saint Cloud
France

1 - INTRODUCTION

Les plus récents systèmes avioniques, par le grand nombre de fonctionnalités qu'ils offrent et par l'utilisation importante de nouvelles technologies dans leur réalisation ont atteint un haut niveau de complexité.

Pour en conserver la maîtrise, l'utilisation d'outils informatiques assurant la mise en oeuvre d'une méthodologie rigoureuse est nécessaire.

D'importants travaux ont été entrepris aux AVIONS MARCEL DASSAULT - BREGUET AVIATION, en collaboration avec d'autres industriels et la participation du Ministère de la Défense, pour définir et développer un ensemble d'outillages informatiques homogènes soutenant les différentes phases de développement de systèmes avioniques.

L'objet du présent docuement est de décrire la nature des besoins au regard du cycle de vie du développement des systèmes avioniques, puis les principaux travaux engagés pour assurer la couverture des différentes phases.

2 - SYSTEME AVIONIQUE

2.1 - OBJET D'UN SYSTEME AVIONIQUE

Les principales fonctions assurées par un système avionique concernent :

- le dialogue avec l'équipage,
- les fonctions opérationnelles (navigation, conduite de tir d'armes, reconnaissance, ...),
- la préparation et la restitution de mission,
- la coopération entre avions et le sol,
- la maintenance opérationnelle.

2.2 - CARACTERISTIQUES D'UN SYSTEME AVIONIQUE

Les systèmes avioniques actuels sont principalement :

- des systèmes très intégrés pour répondre aux exigences suivantes :
  . assurer un grand nombre de fonctions opérationnelles concourant ensemble à la réussite de la mission,
  . tenir dans un volume restreint et pour la masse la plus réduite possible,

. garantir un bon niveau de dialogue avec le pilote tout en cherchant à diminuer sa charge de travail,

Cette intégration se traduit essentiellement par une interdépendance des fonctions et un multiplexage des ressources capteurs, commandes et visualisations.

- des systèmes *fortement évolutifs pour* :

  . pouvoir subir des transformations liées à de nouvelles technologies de matériel,

  . *être capables de supporter l'intégration de fonctions* pour répondre à de nouveaux besoins opérationnels,

  . permettre des améliorations de la mission,

  . satisfaire à des demandes spécifiques d'un client par le développement d'une version particulière.

- des systèmes *hautement sécurisés pour* :

  . assurer la sécurité de l'équipage par une gestion de pannes intégrée (surveillance, détection, reconfiguration).

Les progrès technologiques de ces dernières années ont permis une forte numérisation des systèmes avioniques ; cette numérisation s'est concrétisée par la multiplication de processeurs spécialisés dans les équipements, l'utilisation de bus numériques et un volume croissant des logiciels.

Tous ces éléments réunis ont favorisé la réalisation de systèmes de plus en plus intégrés, performants et évolutifs, mais dont le niveau de complexité, lié au grand volume de logiciels et à la combinatoire des logiques systèmes, *a rendu la maîtrise difficile*.

2.3 - PROBLEMES POSES

Les problèmes posés lors du développement des récents systèmes avioniques concernent particulièrement :

- la maîtrise des coûts et des délais,

- la communication entre le grand nombre de partenaires industriels impliqués dans un programme,

- la validation des systèmes et de leurs évolutions,

- la gestion d'une importante documentation de spécifications et les difficultés de sa mise à jour.

Les chiffres cités ci-dessous sont issus du développement d'un système avionique type MIRAGE 2000 et illustrent ces derniers points.

Pour assurer l'ensemble des fonctions de "base" dites de conduite de la machine (conditionnement, génération électrique, pilotage, navigation, localisation, ...) et des fonctions spécifiques missions (contre-mesures, interception air/air, attaques air/sol, reconnaissance, ...) le système avionique conduit à l'intégration d'une centaine d'équipements et à la réalisation de 500 Kmots de logiciels embarqués.

Rapportés aux différentes phases du cycle de vie, les volumes de documentations associées sont :

- *définition et spécification système*

  . l'ensemble de la documentation de spécification système représente une hauteur de l'ordre de 3 mètres.

- conception et réalisation du logiciel

  . à 500 Kmots de logiciels temps réel, la documentation de l'analyse-conception et de *maintenance correspond* à 2,5 mètres dont 2 mètres de listing source.

- validation, intégration système :

  . pour la *validation d'une fonction opérationnelle*, la documentation associée représente 1 mètre.

La difficulté de mise au point de ces systèmes et l'adéquation du produit final réalisé au vu de l'expression des besoins opérationnels se traduisent par la rédaction moyenne de 3,5 fiches de modification par jour, ces fiches correspondant à des demandes de correction ou d'évolution.

Pour le développement d'un système avionique, le temps passé sur chacune des étapes, exprimé en pourcentage de la durée totale est de l'ordre de :

- phase de définition, spécification système	30%
- phase de réalisation, validation équipement	50 %
- phase d'intégration au sol	20 %
- phase d'essais en vol	10 %

3 - OBJECTIFS DE L'UTILISATION D'OUTILLAGES INFORMATIQUES

La complexité du développement des sytèmes avioniques justifie pleinement l'emploi systématique d'outils informatiques adaptés pour supporter l'ensemble des activités du cycle de vie.

Cet ensemble d'outils comprend des "outils de fond" pour permettre l'acquisition de connaissances sur les fonctions et systèmes à développer, et des outils formels dont le rôle essentiel est de fournir une aide à l'application d'une méthodologie rigoureuse.

Par l'utilisation d'un ensemble d'outils informatiques adaptés aux différentes étapes du développement, les objectifs poursuivis sont les suivants :

- homogénéïté des méthodes de travail par l'utilisation d'outils communs,
- aide à la définition de systèmes intégrés et évolutifs,
- obtention de spécifications de qualité,
- amélioration des communications entre les différents partenaires par la structuration et la formalisation des échanges,
- aide à l'élaboration de la documentation,
- réduction des coûts et des délais par :
  - . la réutilisation issue de la capitalisation de l'expérience,
  - . une meilleure productivité,
  - . une minimisation des phases de validation/intégration lors des essais au sol et en vol par l'amélioration de la qualité des spécifications et leur validation avant réalisation.

Il faut noter ici, que le coefficient d'amplification des coûts, pour une erreur de conception ou de spécification qui n'est découverte qu'aux essais au sol ou en vol, est de l'ordre de 50 à 100.

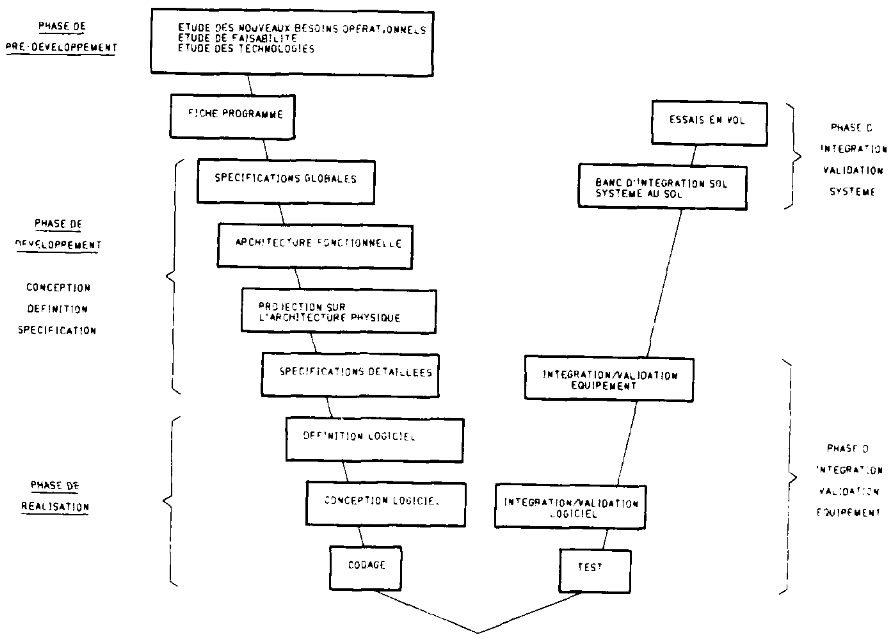
4 - LE CYCLE DE VIE D'UN DEVELOPPEMENT DE SYSTEME AVIONIQUE

Au cours de cette présentation, l'on se propose de détailler les différentes étapes du cycle de vie d'un développement de système avionique et d'exprimer les besoins associés.

4.1 - PRINCIPALES PHASES DU CYCLE DE VIE

Au vu de leurs expériences en matière de développement de systèmes, et pour répondre aux exigences d'un système avionique, les AMD-BA se sont définis une méthodologie de développement illustrée par le cycle de vie ci-après.

CYCLE DE VIE DE DEVELOPPEMENT DES SYSTEMES AVIONIQUES



4.2 - PHASE DE PRE-DEVELOPPEMENT

4.2.1 - Définition

Lors de cette phase, les buts poursuivis sont principalement :

- l'acquisition de la connaissance des moyens et du savoir-faire, c'est-à-dire le parcours sur la possibilité de mise en oeuvre de nouveaux apports technologiques,
- L'évaluation de la faisabilité de nouveaux concepts de fonctions opérationnelles.

Les travaux menés lors de cette phase sont assimilables à des travaux de recherche opérationnelle, d'études générales, de réalisation de maquettes probatoires pouvant donner lieu à des simulations partielles ; l'objectif de ces travaux est de réunir les éléments qui permettront de définir des meilleurs projets de systèmes et d'établir les plans de développement techniques et industriels.

Cette phase fait appel à de nombreux interlocuteurs se traduisant notamment par :

- des liens industriels-équipementiers pour les études, la technologie de base et la réalisation de maquettes probatoires,

- des liens industriels-avionneurs-équipementiers pour l'analyse des contraintes d'avionnage, d'intégration système et la caractérisation des interfaces de dialogue équipage-système,

- des liens avec les centres étatiques pour l'évaluation par des utilisateurs.

4.2.2 - Besoins

Les besoins nécessaires dans cette phase de pré-développement sont essentiellement des "outils de fond" permettant d'acquérir une connaissance ; ces outils reposent sur des :

- moyens de simulation légers ou de maquettage,

- moyens de simulation lourds.

4.3 - PHASE DE DEFINITION

4.3.1 - Spécifications globales

4.3.1.1 - Définition

A partir des résultats de la phase de pré-développement (études de faisabilité et pré-études de performances) et des besoins opérationnels exprimés par la fiche programme, cette phase a pour objectif la définition préliminaire du système.

Par l'analyse des différentes missions dont sera capable l'avion, il est procédé au recensement des fonctions opérationnelles et des moyens nécessaires. Cette étape fait une première description des fonctions opérationnelles, du poste de pilotage, des capteurs et de l'architecture matérielle sous la forme de spécifications générales fixant le cadre des travaux à réaliser.

A l'issue de ces travaux, il est réalisé un document "modes et commandes" fournissant l'enveloppe des possibilités du système avionique. Ce document comporte deux parties orientées "matériel" et "fonctionnel" :

- la partie "matériel" présente les équipements ou sous-ensembles du système avionique, ces derniers désignant les capteurs complexes possédant différents modes de fonctionnement tels que radar, capteurs optroniques, ...,

- la partie "fonctionnel" présente une philosophie générale d'utilisation de la cabine (commandes et visualisations) et décrit les différents modes de fonctionnement du système.

A partir de ce document "modes et commandes", chaque fonction opérationnelle fait l'objet d'un document "spécification globale" traduisant, en termes d'objectifs à atteindre, le scénario opérationnel d'utilisation : les moyens d'activation, les principales commandes/actions possibles, les visualisations et les moyens de sortie de la fonction.

4.3.1.2 - Besoins

Les besoins ressentis pour couvrir cette étape sont de deux natures :

- moyen de validation des concepts de dialogue Homme-machine et de certains algorithmes.

  Cette validation devra recourir à des outils légers de simulation mais offrant une restitution suffisante de l'environnement cabine pour permettre un rebouclage avec des utilisateurs finaux.

- moyen de formalisation des résultats des travaux pour les relectures puis le travail aval de conception.

Pour faciliter les relectures documentaires effectuées par un grand nombre d'intervenants, la formalisation doit conserver l'utilisation d'un langage naturel mais garantir l'homogénéité des documents par le respect de plan-types.

4.3.2 - Architecture fonctionnelle

4.3.2.1 - Définition

A partir des besoins exprimés lors de l'analyse fine des scénarios d'utilisation des fonctions opérationnelles, des hypothèses de mise en oeuvre des capteurs et de la philosophie d'utilisation de la cabine, cette étape a pour but la description de l'architecture fonctionnelle du système. Le résultat attendu est un découpage du système en différents modules fonctionnels prenant en compte des règles d'évolutivité, de modularité, de récupérabilité et, résolvant les conflits de ressources capteurs, commandes, visualisations liés à la superposition des fonctions.

Cette architecture met en évidence :

- les chaînes fonctionnelles par le cheminement des interfaces entres les différents modules,
- l'interdépendance des fonctions,
- le fractionnement du système en modules fonctionnels pour lesquels les interfaces entrées-sorties sont identifiés.

4.3.2.2 - Besoins

Cette étape, essentielle pour la maîtrise des systèmes et particulièrement pour la maîtrise des évolutions et des modifications, justifie le recours à l'utilisation d'un outil formel pour garantir la visibilité du système. Le formalisme recherché doit permettre :

1 - Une aide à la conception du système en soutenant la démarche de découpage,

2 - Une lisibilité fonctionnelle de chacune des fonctions opérationnelles au sein de l'architecture complète.

3 - La facilité d'exploitation des résultats pour les besoins des étapes avales.

4 - L'amélioration de la communication entre les différents concepteurs impliqués dans la définition de l'architecture fonctionnelle.

4.3.3 - Architecture physique

4.3.3.1 - Définition

Le résultat de la précédente étape permet de disposer d'une connaissance du système sous la forme d'un ensemble de modules fonctionnels et de leurs interfaces. L'objet de cette étape est de projeter cette architecture fonctionnelle sur l'architecture matérielle définie lors de la phase préliminaire et, par itérations successives, de trouver la meilleure répartition.

Ainsi les différents modules fonctionnels sont répartis dans les différents équipements, ce qui permet d'identifier à ce stade la nature de transmission des échanges : analogiques ou numériques.

L'affectation de ces modules aux différents équipements, outre les évidentes affectations au vu des fonctions premières des équipements, s'effectue au regard des critères suivants :

- évolutivité, charge calcul et volume mémoire des processeurs, débits des échanges transitant sur les bus, ...

4.3.3.2 - Besoins

Au cours de cette étape de superposition de l'architecture fonctionnelle sur l'architecteure matérielle, un très grand nombre d'informations d'interfaces est manipulé et cette manipulation est reconduite lors des multiples itérations. Aussi le besoin se fait ressentir de disposer pour ce stade de définition d'un outil de gestion doté de mécanismes visant à automatiser certaines opérations. De plus, il convient de pouvoir effectuer une validation de l'architecture physique au sens des transmissions numériques par l'évaluation des charges bus.

4.3.4 - Spécifications détaillées

4.3.4.1 - Définition

Sur la base des résultats des étapes amont, est établi un ensemble de spécifications qui donneront lieu à la réalisation des équipements et de leurs logiciels.

Chaque équipement est défini par les spécifications suivantes :

- Clauses Techniques d'Intégration (CTI) :

  . Son objet est de définir l'implantation dans l'avion, les contraintes d'avionnage, le rôle de l'équipement et son intégration au sein du système.

- Spécifications Détaillées de Fonctions Opérationnelles (SDFO) :

  . Ces documents spécifient les traitements effectués par les fonctions logicielles dans l'équipement ; pour les équipements de visualisations, ces spécifications définissent, de plus, la symbologie présentée au pilote.

- Fiche d'Interface Analogique (FIA) :

  . Pour chaque liaison analogique de l'équipement, ces fiches spécifient la nature du câblage et les interfaces électroniques d'émission ou de réception.

- Fiche d'Interface Digibus (FID) :

  . Ces fiches spécifient l'ensemble des communications (échange, fréquence, nature, codage, ...) de l'équipement via le(s) bus avec le reste du système.

4.3.4.2 - Besoins

L'ensemble des produits de cette étape forme le noeud de communication entre l'avionneur-systèmier et les équipementiers. Aussi le besoin essentiel est d'établir la meilleure communication possible entre tous les intervenants. Cette recherche de la meilleure communication conduit à des formalismes adaptés à chacune des spécifications et à l'utilisation d'aides pour éliminer les ambiguïtés, les omissions, les incohérences ... .

Pour accroître l'efficacité et limiter les découvertes tardives d'anomalies et des erreurs de spécifications, le besoin s'exprime en termes de maquettage.

Les objectifs du maquettage sont les suivants :

- s'assurer que le besoin est satisfait,

- donner aux concepteurs des moyens de vérification par "une animation des spécifications papier" pour découvrir très tôt les anomalies fonctionnelles ou les erreurs de spécifications,

- élaborer des jeux de mise au point qui pourront être réutilisés lors des premières validations des logiciels réels,

- tester et valider les évolutions.

Il faut noter la limite de cette validation qui ne peut prendre en compte les ressources calculateurs et bus qu'avec une notion imparfaite du temps ; il s'agit essentiellement d'une validation fonctionnelle des spécifications.

4.4 - PHASE DE REALISATION

4.4.1 - Définition, Besoins

L'ensemble des spécifications issues des phases précédentes, forme le point d'entrée de la phase de réalisation des équipements et des logiciels.

Pour répondre aux exigences croissantes en terme de criticité, de volume, de multiplicité des versions du logiciel, les efforts ont porté sur la méthodologie, le gain en productivité, l'automatisation du processus de développement ; en particulier, la phase de réalisation est subdivisée en sous-phases où activités formalisées, et pouvant être validées avant le passage à la phase suivante.

Aussi pour répondre à ces objectifs, il convient de supporter l'ensemble des étapes de réalisation des logiciels par un ensemble d'outils cohérents et adaptés ; ces étapes sont :

- Définition de logiciel.
- Conception de logiciel.
- Codage sur des chaînes de programmation utilisant plusieurs langages pour répondre au mieux aux contraintes des différents programmes opérationnels.
- Tests statiques et dynamiques.

4.5 - PHASE D'INTEGRATION ET DE MISE AU POINT

Après la réalisation des équipements et leur validation unitaire chez les différents industriels, la notion de système va devenir effective. La première étape de validation, à ce stade, concerne les travaux d'intégration effectués sur les "bancs d'intégration système".

Un banc d'intégration système se définit comme un moyen de stimulation, à partir d'enregistrement de vols réels, de l'ensemble des équipements du système câblés sur le câblage réel de l'avion.

Le but de ces bancs d'intégration est la validation et la mise au point du système avionique et donc de ses fonctions opérationnelles, dans un environnement et une configuration réalistes, aux fins de limiter le nombre des essais en vol.

4.6 - ESSAIS EN VOL

Les essais en vol sont la phase finale de validation des systèmes avioniques.

4.7 - L'ASPECT DES EVOLUTIONS

4.7.1 - Origine

Les demandes d'évolution trouvent leur origine à deux niveaux :

- les évolutions demandées pour que le système satisfasse au mieux aux besoins opérationnels initialement exprimés dans la fiche programme : ce sont les demandes de modifications et d'améliorations issues des phases de mise au point et de validation sol, vol.

- les évolutions demandées pour que le système satisfasse à de nouveaux besoins opérationnels ; ce sont les demandes d'évolutions correspondant à de nouvelles demandes clients.

Dans les deux cas, le traitement sera identique, comme indiqué ci-après :

4.7.2 - Procédure

Toute demande ou proposition d'évolution se traduit par l'écriture d'une fiche de modification établie en termes opérationnels au niveau système.

Cette fiche de modification système est injectée dans le cycle de vie de développement au niveau approprié pour y être instruite ; cette instruction permet d'établir l'ensemble des fiches d'évolution pour chacun des produits constituant le système et touchés par cette évolution.

Une fois instruites, ces fiches de modifications système sont proposées lors de conférences techniques qui, au regard des coûts, des délais et de l'intérêt opérationnel, décident de leur application.
Lors du développement, c'est par le biais de ces conférences techniques que le système avionique évolue en différents "états système de développement" jusqu'à l'état opérationnel appelé "standard". De façon identique, le système opérationnel évolue de standard en standard pour répondre à de nouveaux besoins clients.

4.7.3 - Besoins

Au regard des évolutions, le besoin s'exprime par :

- gérer de façon rigoureuse les demandes d'évolution et leur suivi,

- maintenir à jour l'ensemble de la documentation de spécification,

- être capable de consulter la définition de chaque "état système".

5 - TRAVAUX ITI

5.1 - PRESENTATION

Dans le cadre du cycle de vie de développement d'un système avionique et du fait de l'accroissement naturel des relations entre tous les coopérants participant au développement de système (avionneurs, équipementiers, centre d'essais en vol, Ministère de la Défense, ...) il est apparu nécessaire de mieux organiser et renforcer ces relations plus particulièrement pour tout ce qui concourt à la conception du système, à la production et à la maintenance des logiciels.

L'étude ITI (Intégration et Traitement de l'Information) rassemble des industriels avionneurs (AEROSPATIALE et AVIONS MARCEL DASSAULT - BREGUET AVIATION) et équipementiers (CROUZET, ELECTRONIQUE SERGE DASSAULT, SAGEM, SFIM, SFENA, THOMSON-CSF) déjà entraînés à coopérer depuis de nombreuses années pour réaliser des systèmes avioniques, ce qui permet de proposer de façon commune la réalisation d'un système de développement d'avionique (SDA) qui permette en particulier de formaliser les relations entre les industriels coopérants permettant ainsi d'assurer une meilleure qualité à un coût moindre et de profiter des compétences et expériences acquises.

5.2 - HISTORIQUE

Au cours de l'année 1983, sur la constation de l'apparition systématique des logiciels dans les équipements et l'intégration de plus en plus grande des différentes fonctions, l'étude ITI a orienté ses travaux vers la réalisation d'un système de développement avionique. Cette orientation répondait au besoin de maîtriser les délais et assurer la qualité des systèmes et plus particulièrement le développement des logiciels. Le logiciel, par son interchangeabilité et son coût de production en série quasiment nul, a permis de reporter sur lui des modifications qui auraient été longues et coûteuses d'introduire dans la partie matérielle. Cependant, après quelques années d'expérience, les industriels aéronautiques se sont aperçus que cette souplesse du logiciel et son introduction massive mal maîtrisée pourraient présenter des risques graves.

Les premiers travaux engagés ont permis d'établir un cadre méthodologique commun, synthèse d'un questionnaire présentant les méthodes et outils de chacune des sociétés. Un cycle de vie standard a été établi mettant en évidence chacune des phases et les activités associées. Pour chaque phase, l'analyse en matière d'outillage a été conduite sous les angles :

- offrir au concepteur-développeur une aide à la nature de l'activité qu'il exerce,

- offrir des moyens de communication.

Ces travaux se sont concrétisés début 85 par un document de spécifications globales du SDA.

A l'issue de ces travaux finalisant l'expression des besoins et à partir de l'évaluation d'un grand nombre d'outillages existants, la solution retenue a conduit à la définition de 2 ateliers intégrés : Un Atelier Système et un Atelier Logiciel communiquant entre eux.

L'Atelier Système couvre les phases de conception et de définition de système, l'Atelier Logiciel les phases de conception et de développement logiciel.

D'autre part, l'atelier ne gère pas seulement des documents mais aussi les informations contenues dans ces documents ; l'atelier est caractérisé par un grand formalisme des outils permettant d'accéder aux données manipulées entre outils et la base de données de référence. L'atelier intégré a l'avantage de fournir une vision globale et cohérente du projet, une réduction du volume d'informations avec une plus grande fiabilité de ces informations et une automatisation des contrôles.

La solution retenue a fait l'objet de la rédaction d'une Spécification Détaillée du SDA fin 85.

Il faut noter que la particularité de ces ateliers est d'être "ouverts", au sens où chaque industriel peut compléter et intégrer ses outils propres afin d'assurer une meilleure couverture du cycle de vie.

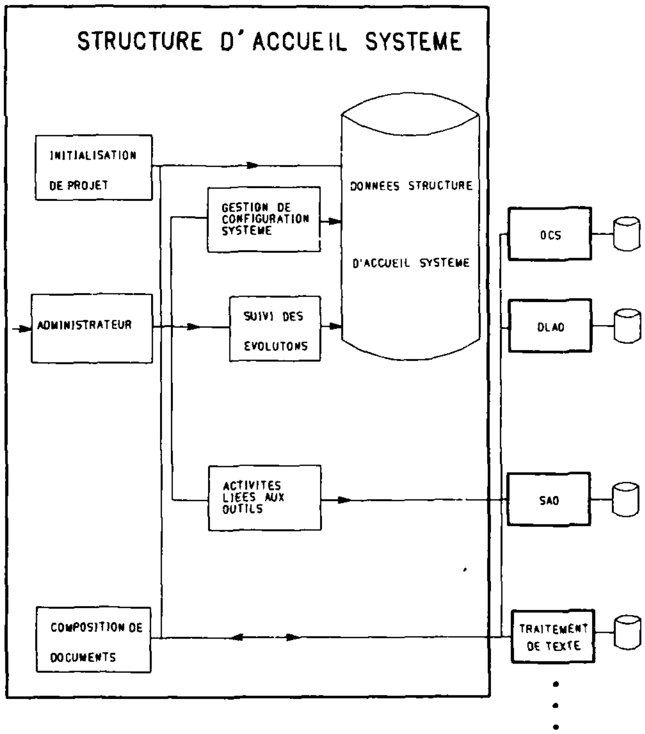
6 - SDA ITI

Nous nous proposons ci-dessous d'introduire la définition des Ateliers Système et Logiciel ; une présentation plus précise faisant l'objet d'une seconde présentation : ATELIER DE CONCEPTION DE SYSTEME AVIONIQUE ET DE REALISATION DE LOGICIEL EMBARQUE.

6.1 - ATELIER SYSTEME

Il s'agit de l'ensemble des outillages mis en place permettant à l'Atelier Système de gérer :

- la logique de développement du système,
- les dialogues avec tous les intervenants,
- les demandes d'évolution d'un système en développement,
- les activités de conception et de définition système supportées par des outils adaptés.



6.1.1 - Structure d'Accueil système

La Structure d'Accueil Système assure les fonctions de :

- initialisation de projet
- suivi des évolutions
- gestion de configuration système
- intégration des outils spécifiques de l'atelier
- administrateur gérant les droits d'accès.

Plus particulièrement son rôle a pour objet :

- l'intégration des différents outils de l'Atelier Système par des échanges standards d'informations entre un outil amont et un outil aval,
- la gestion de l'ensemble de la documentation de spécification du système,
- la gestion du plan de développement,
- la gestion des communications entre les différents partenaires,
- par l'apport d'un outil de documentation, elle assure la composition des résultats des différents outils pour l'élaboration de la documentation de spécification du système,
- la gestion du suivi des évolutions et la garantie de la cohérence de l'ensemble des produits de spécification issus des outils par la gestion de l'application de mécanismes au niveau de chacun des outils de l'Atelier assurant :
    - . l'instruction de la modification sous l'outil tout en conservant l'état "avant" de la modification,
    - . la mesure de l'impact de la modification par le contrôle de la modification en elle-même et vis-à-vis de la spécification,
    - . la rédaction automatique de la fiche d'évolution.

6.1.2 - Outil de Conception Système (OCS)

La phase de définition est couverte par un outil en cours de développement OCS (Outil de Conception Système).

Il supporte graphiquement une méthode d'analyse hiérarchique, structurée et descendante permettant d'appréhender des systèmes complexes de manière générale et d'en découvrir les détails au cours de l'étude.

Cet outil reprend un certain nombre de concepts de la méthode IDEFO auxquels ont été adjoints de nouveaux concepts pour :

- aider à la définition d'interfaces effectués simultanément par plusieurs concepteurs,
- identifier les différentes chaînes fonctionnelles,
- apporter des mécanismes de récupérabilité.

6.1.3 - Outil commande de définition assisté par ordinateur (DLAO)

La phase de spécification détaillée est couverte par un outil en cours de développement DLAO (Définition de Logiciel Assistée par Ordinateur). Cet outil, utilisé pour l'élaboration de spécifications de logiciel temps réel, permet :

- d'accroître la qualité des spécifications par un langage formel associé à des contrôles afin de limiter les erreurs de type ambiguïté, omission, incohérence, surspécification, ...,
- d'améliorer la communication par la production de documents fiables et adaptés aux besoins des différents intervenants.

6.1.4 - Un outil de spécification assisté par ordinateur (SAO)

En phase de spécification détaillée, un outil complémentaire SAO (Spécification Assistée par Ordinateur) est utilisé.

Utilisant un outil de CAO schématique, cet outil permet la spécification sous la forme de planches graphiques des liaisons inter-équipements sans préjuger de la réalisation (matériel ou logiciel) ; ces planches sont utilisées comme support de communication entre les différents partenaires.

7 - ATELIER LOGICIEL

Les solutions de l'Atelier Logiciel sont proches de l'Atelier Système en ce qui concerne les fonctionnalités : l'on trouvera une Structure d'Accueil assurant la gestion de configuration, le suivi des évolutions, l'intégration des outils, le rôle d'administrateur et des outils spécifiques pour couvrir les étapes de définition logiciel, conception logiciel, différentes chaînes de programmation et des outillages de test.

Ce projet ENTREPRISE d'atelier logiciel, developpé sous la tutelle du Ministère de la Défense, est décrit dans la seconde présentation.

8 - UTILISATION DE L'ATELIER SYSTEME AUX AMD/BA

Le coeur de l'Atelier Système AMD-BA est constitué de l'Atelier Système ITI et de l'ensemble des outillages communs auxquels ont été adjoints un certain nombre d'outils propres de manière à assurer une meilleure couverture de certaines étapes du cycle de vie.

Nous présenterons ci-dessous les outils complémentaires puis un schéma illustrant le support apporté par l'ensemble des outils utilisés aux AMD-BA au cycle de vie complet de développement.

8.1 - OUTILS SPECIFIQUES AMD-BA

8.1.1 - Outil d'étude amont

La phase de spécification globale, s'appuie sur un outil informatique OASIS (Outil d'Aide à la Spécification d'Interface Homme-Système).

Cet outil permet la simulation pilotée de fonction opérationnelle, représentative des commandes et visualisation de la cabine, pour l'étude et la validation du dialogue Homme-machine et de certains algorithmes.

Très souple d'utilisation, facile de mise en oeuvre et rapide pour l'exécution de modifications, cet outil d'étude offre un rebouclage rapide et efficace entre les concepteurs et les utilisateurs et ce, dès le niveau des premières définitions.

8.1.2 - Outils formels complémentaires

Phase architecture physique

Un outil spécifique OEA (Outil d'Etude d'Architecture), en cours de réalisation, permet de courvir cette étape. Il assure une validation, au sens des transmissions numériques, de l'architecture ainsi définie par l'évaluation des charges bus.

Phase spécification détaillée

- Un outil de spécification d'imageries avioniques MITIA (Moyens Informatiques pour le Traitement des Imageries Avioniques) :

  . Cet outil se présente dans un environnement de CAO (Conception Assistée par Ordinateur) dont un ensemble de fonctions est particularisé et enrichi pour cette utilisation ; il permet de spécifier par création ou modification des réticules ou des figurations complètes.

  . Le résultat issu de cet outil présente la spécification sous la forme d'édition papier ou d'un fichier informatique dit "neutre" (fichier indépendant des machines graphiques) ; ce fichier par son formalisme et sa "neutralité", tisse un réseau permettant une communication aisée entre les concepteurs, les centres de simulation et les équipementiers chargés de réaliser le logiciel de visualisation.

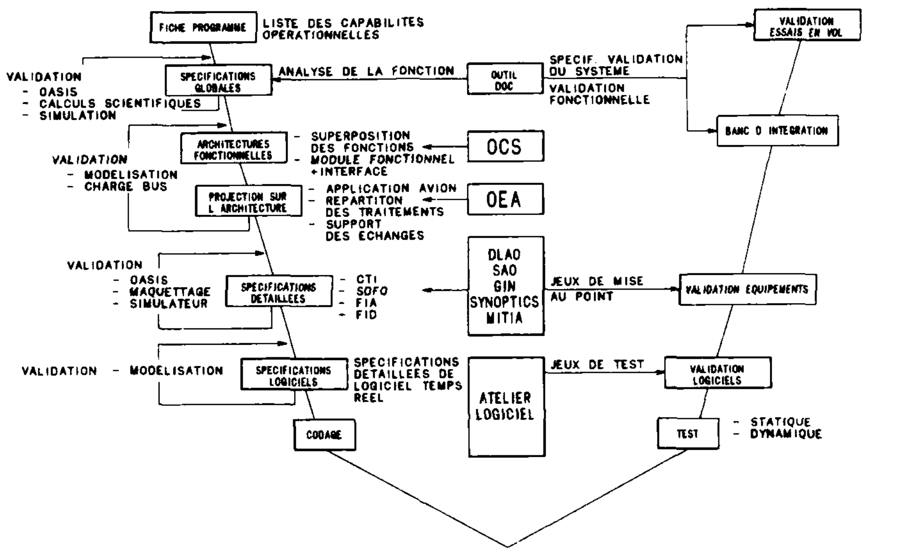
- *Un outil de spécification du câblage avion et des fiches d'interfaces analogiques* SYNOPTICS :
  . Bâti autour d'un logiciel de CAO schématique électrique, cet outil permet la saisie des plans synoptiques de câblage avion et la réalisation des fiches d'interfaces analogiques ; il vérifie les liaisons électriques au regard des règles de l'art et contrôle la cohérence de l'ensemble de la liasse schématique.

- *Un outil de spécification des fiches d'interfaces numériques* GIN (Gestion des Interfaces Numériques) :
  . Cet outil supporte la spécification des échanges entre les différents équipements selon l'architecture numérique composée d'un ou plusieurs bus et, à partir des procédures de gestion d'échanges de type GINA, 1553-B, ARINC.

8.2 - COUVERTURE DU CYCLE DE VIE

Au regard du cycle de vie du développement d'un système avionique, la couverture de l'Atelier Système précèdemment exposée se synthètise par le schéma suivant :

ATELIER SYSTEME DANS LE CYCLE DE VIE



9 - EXPERIENCES PARTIELLES ; CONCLUSION

Un certain nombre d'outillages informatiques, préfigurant cet Atelier Système, ont déjà été utilisés pour le développement de nos récents systèmes avioniques. Leur utilisation a permis de nous conforter dans la définition de l'Atelier Système par l'apport de gains tels que :

- l'amélioration des communications entre les partenaires industriels,

- la meilleure qualité des spécifications,

- l'aide apportée au suivi de la définition du système.

De plus, ces expériences ont mis l'accent sur :

- la recherche de la meilleure convivialité de ces outils pour améliorer la productivité,

- le besoin d'intégration de ces outils entre eux,

- l'aide apportée par ces outils par la mise à jour de la documentation des spécifications.

L'ensemble de cette démarche est analoque à celle qui a été menée en matière de CAO-CFAO ces dernières années et qui a conduit à la réalisation de grands produits logiciels tel le logiciel CATIA (Conception Assistée Tridimensionnelle Inter-Active), réalisé aux AMD-BA.

Convaincus au sein de la communauté avionique française du besoin de disposer d'un Atelier Système pour le développement des nouveaux programmes, nous avons lancé le développement de cet atelier en vue d'aboutir à sa réalisation au début 89.

A SOFTWARE LIFE CYCLE SUPPORT ENVIRONMENT

by

L.Y. Bajwa
W.R. Wisehart
General Research Corporation
P.O. Box 6770
Santa Barbara, CA 93160-6770
U.S.A.

Frank S. LaMonica
Rome Air Development Center
Griffiss Air Force Base, N.Y. 13441
U.S.A.

SUMMARY

Under sponsorship of the U.S. Air Force Rome Air Development Center (RADC), a Software Engineering Environment known as the *Software Life Cycle Support Environment* (SLCSE) was specified and is currently undergoing a 24 month advanced development.

The *SLCSE* is a distributed, computer-based environment of software development tools and methods which span the full spectrum of the software life cycle and are integrated through a common and consistent user interface and a life cycle project database. Primary features of the SLCSE are: (1) it supports both the development and management of complex mission-critical computer resources (MCCR) software systems in accordance with the DoD-STD-2167 life cycle model for software development, (2) it is multi-lingual - supporting the Air Force standard higher-order languages including MIL-STD-1589B (JOVIAL J73) and MIL-STD-1815A (Ada), and (3) through its unifying framework, it enables the integration of both newly developed and existing (off-the-shelf) software tools. This paper will present the SLCSE development effort, including highlights of the SLCSE's architecture, requirements and top-level design. The implementation approach, which consists of a series of eight incremental builds, will be described along with the results of the first build. Testing and verification plans will be discussed and risk areas identified. Finally, long-term plans by RADC to evolve the SLCSE beyond the current contractual requirements will be indicated.

1 INTRODUCTION

The Software Life Cycle Support Environment (SLCSE) is a collection of integrated tools operating on a Digital Equipment Corporation VAX/VMS computer that will support the production of MCCR software throughout its entire life cycle. The tools are embedded in a framework consisting of a user interface and a database. At the highest level, integration of tools is brought about by this framework, which provides a commmon "look and feel" for all of the tools and provides access to all data available to a SLCSE user.

Supplementing this level of integration are the user assistance features, which include tailorable, knowledge-based software development methodology direction, in addition to the usual on-line information regarding details of tool operations. Finally, at the detail level, the tools are integrated in a particular instance of the environment by the specific management and development methods that are incorporated into the knowledge-base and the specific database entities that are incorporated int the overall database schema. These data inputs are tailored to the project and project management parameters such as the life cycle model of the project, the project size, the skills and experience of project personnel, and so on.

1.1 Features

The principal features of SLCSE that address the needs of MCCR software development across its life cycle (regardless of the life cycle model chosen) are as follows:

Completely Integrated Toolset. The project's needs for the entire life cycle are met by integrating a complete toolset in support of critical life cycle activities. Additionally, individual users will be identified by the role they are currently playing and directed to the proper tools and action sequences for performing role functions.

Flexible Development Methodology. The SLCSE will allow software development methodology to be specified by the project manager. Consequently the individual user roles can be delineated in detail, including differing levels of control for different roles. If the specified methodology permits a certain amount of trial-and-error development for certain user roles, SLCSE can be instructed not to issue error messages if a set of tools is not used in a particular order. However, if a well-defined set of procedures is to be followed by another class of users, SLCSE can be set up to enforce those procedures exactly as defined.

Development and Management Support. The needs of MCCR software project managers will be addressed by the SLCSE. Project management planning, scheduling, and costing aids as well as problem reporting and status monitoring tools will be supplied. Configuration management tools will also be in place to support both software developers

and managers. As the repository of all project information, the project database will provide management with data for tracking the development that is usually difficult or impossible to gather.

Support for Automated Document Preparation. A major problem of life cycle support is maintenance of documentation, making it consistent with the current software product and not several versions behind. The concept of enforceable methodology (including the enforced documentation of changes), the total availability of all project information in a common database, and the use of document generation and updating tools will permit SLCSE to provide complete and timely documentation throughout the life cycle.

1.2 Development Team

General Research Corporation (GRC) is the prime contractor for developing the SLCSE. Subcontractors are Intermetrics, Inc. of Cambridge, Massachusetts, and Software Productivity Solutions, Inc. of Melbourne, Florida. Development is funded by the US Air Force Systems Command through the Rome Air Development Center (RADC). The design is based on a definition study previously sponsored by RADC and performed by GRC teamed with Intermetrics.

2 SLCSE ARCHITECTURE

2.1 Project Objectives

The long-term development of SLCSE will be an evolutionary process. The current contract will produce a series of eight builds, two of which will be delivered to RADC during the course of the project as intermediate "proof of concept" versions. The final build, SLCSE 3.0, will be delivered as the end product of the project, and will have a toolset capable of supporting the full DoD-STD-2167 life cycle. More importantly, however, this version will demonstrate the integrated environment principle, producing a framework for the continued evolution of the SLCSE toolset. SLCSE 3.0 will provide a usable life cycle environment, capable of supporting MCCR software projects. It will be complete with respect to the framework and integration concepts that have been identified as essential to meeting the goals of productivity and quality and will offer a respectable catalog of tools.

2.2 Architecture Overview

Figure 2.1 shows the overall SLCSE architecture, and the relationship of the three subsystems.

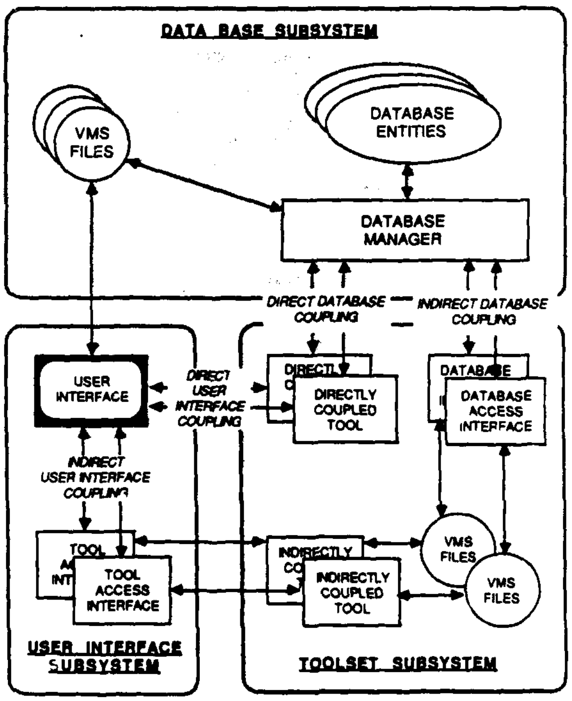


Figure 2-1. SLCSE Subsystems.

User Interface. The primary goal of the user interface is to provide a single shell to the user for communicating with all tools in the SLCSE, for transferring data within the SLCSE, and for getting assistance on the use of tools and the environment itself. The project will produce a standard set of user interface packages that will be supplied to builders of new tools along with guidelines for developing tools that match the desired interface principles. In SLCSE 3.0, invocation of an existing tool will adhere to the SLCSE user interface principles, but once invoked the tool will continue to use its current command language or menu systems. When off-the-shelf tools are updated, they will be reviewed to determine the feasibility of redesigning their user interfaces to the SLCSE standards.

Database. As with the user interface, the database subsystem for SLCSE 3.0 will emphasize the development of packages that will support the long-term goals of a fully integrated toolset. Interface routines will permit existing tools to make use of the SLCSE database, but databases or data files generated by these tools will not be replaced in SLCSE 3.0. New tools, however, will communicate directly with the database, following guidelines published for using the database subsystem utilities. In the future, tools having their own internal databases may be redesigned for use of the SLCSE database utilities, but that is likely to result in a more sweeping redesign than would be required for changing the user interface of an existing tool.

Toolset. Users must perceive SLCSE as a unified collection of tools that work efficiently together. Tools that are added to the environment must be integrated into it either by designing them that way, or by using the facilities of the user interface and database subsystem to provide the integration.

The goal criteria for integrating the toolset are as follows:

(1) Similarly invoked functions must operate from a uniformly similar user interface.

(2) User assistance must be provided in a uniform manner for all tools.

(3) Tools that communicate data to another tool must provide it completely, and in a uniform format that all tool builders will use.

(4) Tools must provide management data in a uniform format to the management tools.

The degree to which these integration goals can be achieved depends on the degree to which tools are "coupled" to the user interface and database utilities.

(1) **Directly coupled tools** will access the user interface and the database by calling their utilities directly to send and receive data, and will adhere strictly to the integration goals.

(2) **Indirectly coupled tools** will generally communicate with the user interface and the database through intermediate files, logical names, and global symbols that are set up by interface routines and command procedures.

It is generally not possible to integrate existing tools under the above criteria. SLCSE 3.0 will call on the framework (i.e., the user interface and the database system) to provide the mechanisms for integrating such tools. This will be accomplished as follows:

(1) Command interpreters and interface routines can be used to meet the top-level user interface requirements, user assistance requirements, and the tool-to-tool communication requirements.

(2) Commands, menus, and help features that are already built into existing interactive tools will be used as is until the maturation of the SLCSE brings about the desired tool modification, or the introduction of new tools tailored to the environment.

The SLCSE 3.0 toolset will be largely assembled from existing tools, drawing on interface routines to provide integration at the tool invocation level. Internal communications with the user and with data files will not be modified. This produces some compromises with respect to long-term toolset integration goals, but no compromises will be made in the framework itself.

Several new tools will be fully integrated into SLCSE 3.0. These include tools built under the SLCSE contract, and the Ada Test and Verification System (ATVS), which was designed under a RADC contract, and is now in its implementation phase.

In contrast to SLCSE 3.0, later versions of SLCSE will strive for direct coupling of all tools. New tools will be built for direct coupling, using standard packages for communicating with the user interface subsystem and database subsystem that will be made available to tool developers as a part of their implementation contract. If an existing tool has a desired capability, it will be modified for direct coupling, or if that is not possible due to the design of the tool, it will be recoded using the standard interface packages. Indirect coupling in later builds will be used primarily for evaluation of an existing tool to determine if its capabilities warrant including it as a regular (and thus directly coupled) component of the environment toolset.

3 SLCSE REQUIREMENTS

The SLCSE is a very complex system and, as such, has a complex set of requirements. However, the top level requirements as set forth in this section are relatively easy to describe.

3.1 User Interface Requirements

Easy to use. The SLCSE must make use of Man-Machine-Interface (MMI) techniques that promote user-friendly operations, such as the use of multiple, overlayed windows and the consistent use of various regions of the screen. Of course, what is user-friendly for the novice might be very frustrating for an individual familiar with SLCSE operations. Therefore, this requirement also dictates that the SLCSE must comfortably support users of all types, providing a spectrum of command alternatives from menu hierarchies to cryptic, keyword phrases.

Supports user roles. The SLCSE framework must support the nineteen user roles depicted in Figure 3-1.

ROLES	FUNCTIONS
Acquisition Manager	VNOQRW
Project Administrator	AVNOQRW
Project Manager	BKVNOPQRW
Project Leader	KVNOPQRW
System Analyst	CDFKNOPQRUW
S/W Analyst	CDEFGKNOPQRVW
Programmer	DEFGNOPQRUW
S/W Test Engineer	CDEFHLNOPQRW
S/W Integrator	HLNOPQRW
S/W Performance Tester	CDEFGHLNOPQRVW
System Integrator/Tester	HLNOPQRW
V&V Personnel	JNOPQRW
QA Representative	INOPQRSW
CM Personnel	FKNOQRSW
PDSS Personnel	MNOPQSW
Training Personnel	ABKNOQRW
MCCR User	KMNOQRW
SLCSE Installation Personnel	NOPQRSTW
Secretarial Personnel	VNOQRW

- A. Project Planning
- B. Project Control
- C. Requirements Specifications and Analysis
- D. Design Specification and Analysis
- E. S/W Prototyping and Modeling
- F. Reusability
- G. Program Generation Testing
- H. Integration Test
- I. QA
- J. Verification and Validation
- K. Configuration Management
- L. S/W H/W Integration
- M. Post-Deployment S/W Support
- N. Communication
- O. Document Generation
- P. Data Collection Performance Measurement
- Q. Help
- R. Training
- S. Transition
- T. Tailoring
- U. Knowledge Engineering
- V. Administration
- W. PR/ECR Reporting

Figure 3-1. SLCSE User Roles.

Supports instantiation of new environments. The purpose of the SLCSE is not to create a single environment, but to provide a means for creating environments to support the unique needs of each individual software development project. This means that a manager should be able to sit down and interactively instantiate a new environment in a matter of hours or days, not weeks or months. This requirement implies that the methods employed must be very straightforward - almost like a cookbook - and that no special knowledge should be required of the creator, other than that already required to exercise a SLCSE environment.

Supports distributed, interprocess control and communications. Many of the SLCSE tools will be standalone, executable images. The SLCSE must be able to "fire off" these tools from the SLCSE image. This requires some sort of interprocess control mechanism.

In addition, interprocess communication facilities are needed to support the distributed nature of SLCSE environments (operating either on a host computer or any number of networked workstations) and "no-wait" requests. "No-wait" requests are generally those that will take a long time to satisfy - like a complicated database query - prompting a user to indicate: "I want this request done, but I don't want to wait for the results."

Supports embedded methods. A life cycle model like DoD-STD-2167 specifies what data and products are to be derived from a software development process. On the other hand, a software development methodology, and the individual methods that support that methodology, specifies how these data and products are to be produced. Standard methods are represented in a SLCSE environment by the capabilities of the tools that make up the environment.

Additional methods may be needed, however, to govern the day-to-day operation of an environment. For example, there should be restrictions on access to tools and there should be rules about the consequences of certain user actions or sequences of actions.

Provides a tool access mechanism. Tools that are indirectly coupled to the SLCSE may require the prior specification of runtime information; for example, the name of a file. The SLCSE must provide a means of collecting these execution parameters, in the form of a tool access mechanism for each SLCSE tool (as required).

Provides information access mechanisms. Information access needs will include file access, an ad-hoc query capability and a database access mechanism. This latter capability is required by indirectly coupled tools, which have no knowledge of external database structures and so must make use of import/export mechanisms. These mechanisms will be used to pull information out of the SLCSE in the form of a file that can be read by the tool (or vice versa).

Provides four kinds of input utilities. Utilities must be provided to assist tool builders in conforming to and making efficient use of the SLCSE user interface. Specifically, utilities will be provided for Menu Input, Forms Input, Matrix Input and Script-Driven Input. Menu Input will support the selection of menu options; Forms Input will support data entry via fill-in-the-blank forms; Matrix Input will support a very simple spreadsheet-like data entry capability; Script-Driven Input will support "dialogue" interactions requiring the data of textual information - like the sections and paragraphs of a document, for example.

Provides four kinds of output utilities. Utilities must also be provided for Display Output, Message Output, Graphics Output and Print Output. Display Output will support "painting" a window with appropriate information displayed within its borders; Message Output will support presentation of prompt, comment and error messages; Graphics Output will support turning raw data into simple scatter and histogram plots; Print Output will support the generation of hardcopy listings with security classification markings.

Incorporates help and training facilities. The SLCSE framework will support two levels of on-line help (brief and detailed) and an on-line training facility.

3.2 Database Requirements

Supports use of a Britton-Lee IDM-500 Database Machine. To help alleviate anticipated database performance problems, the SLCSE must be configured to run with a Britton-Lee IDM-500 Database Machine.

Supports DoD-STD-2167, although not precluding other life cycle models. The U.S. military has in recent years standardized on one particular life cycle model: DoD-STD-2167. The SLCSE is geared toward the support of this model, but will also support other models.

Supports the description of an Entity-Relationship (ER) database model. One way to reduce the complexity of traditional database models is to describe them in terms of entities and relationships like the example shown in Figure 3-2. ER models, because they are more English-like, are much more natural and easier to comprehend. The SLCSE must be able to accept database descriptions - the schema - in terms of an ER model. Additionally, the syntax employed must be one familiar to database model developers and must provide a means for decomposing database structures into smaller structures (i.e. subschemas).

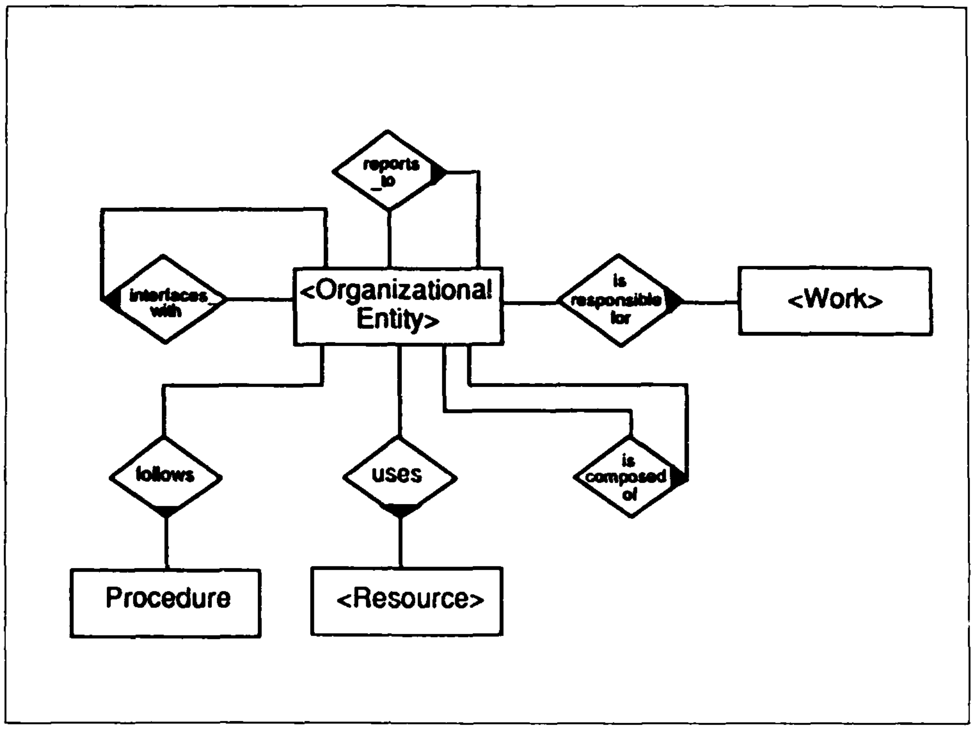


Figure 3-2. Sample Entity Relationship (ER) Model.

Supports ER-type database queries. The SLCSE must support both direct and indirect ER-type database queries. Application tools that "know" about the ER structure will make direct database queries; other applications will depend upon interface mechanisms that make use of a standard query language. Both static (precompiled) and dynamic (on-the-fly) queries must be supported.

Supports version management. The database is not a fixed structure; it has a "timeline" sense as well. The SLCSE must support control and management of different versions of database items.

Supports control of database access. Users must not be allowed to change the same database element at the same time. The SLCSE must support a system of database locking/unlocking mechanisms to prevent such an occurrence.

Additionally, there must be support for data distribution. Although there will be only one master database, hosted by a single computer, the SLCSE must be able to deal with a network of computers running the same environment, but either requesting information from the master database or downloading/uploading portions of the master database.

Supports database archiving. It periodically becomes necessary to archive portions of a database. The SLCSE must support this function in an efficient manner.

3.3 Toolset Requirements

Supports integration of 30 tools in ten tool categories. SLCSE 3.0 will support commercial tools, Government-furnished (GFE) tools and tools especially built for integration into the SLCSE. The tools are listed in Figures 3-3 and 3-4 in terms of ten tool categories. Briefly, the tools are as follows:

TOOL	SOURCE	Included in Build 1	2	3	4	5	6	7	8
General Support Tools									
Text Editor	Part of VMS			■	■	■	■	■	■
Document Formatter	LOP			■	■	■	■	■	■
Command-Language Editor	Part of VMS			■	■	■	■	■	■
Electronic Mail	Part of VMS			■	■	■	■	■	■
Verification Tools									
Tracing Tool	Part of ALICIA							■	■
Code Auditor (Ada)	Part of ATVS							■	■
Static Analyzer									
Ada	Part of ATVS							■	■
JOVIAL J73	Part of J73AVS						■	■	■
FORTRAN	Part of RXVP80				■	■	■	■	■
COBOL	Part of CAVS						■	■	■
Data Flow Analyzer									
Ada	Part of ATVS							■	■
JOVIAL J73	Part of J73AVS						■	■	■
FORTRAN	Part of RXVP80				■	■	■	■	■
COBOL	Part of CAVS						■	■	■
Interface Checker									
Ada	Part of ATVS							■	■
JOVIAL J73	Part of J73AVS						■	■	■
FORTRAN	Part of RXVP80				■	■	■	■	■
COBOL	Part of CAVS						■	■	■
Quality Analyzer									
Ada	Part of AMS							■	■
FORTRAN	Part of AMS							■	■
Requirements Analysis Tools									
Requirements Generator	Unnamed - new tool				■	■	■	■	■
Requirements Documentor	Unnamed - new tool				■	■	■	■	■
Consistency Analyzer	Unnamed - new tool				■	■	■	■	■
Design Tools									
Text-Oriented Design Tool	Part of SDDL				■	■	■	■	■
Design Documentation Tool	Part of SDDL				■	■	■	■	■
Consistency Checker	Part of SDDL				■	■	■	■	■
Coding Tools									
Syntax-Directed Editor									
Ada	From DEC			■	■	■	■	■	■
JOVIAL J73	From DEC			■	■	■	■	■	■
FORTRAN	From DEC			■	■	■	■	■	■
COBOL	From DEC			■	■	■	■	■	■
Compilers									
Ada	From DEC			■	■	■	■	■	■
JOVIAL J73	From PSS			■	■	■	■	■	■
FORTRAN	From DEC			■	■	■	■	■	■
COBOL	From DEC			■	■	■	■	■	■
Assembler	From DEC			■	■	■	■	■	■
Linker	From DEC			■	■	■	■	■	■
Debugger	From DEC			■	■	■	■	■	■
Assertion Translator									
Ada	Part of ATVS							■	■
JOVIAL J73	Part of J73AVS						■	■	■

NOTE: New tools will be developed under the SLCSE contract; partial shading means prototype version of tool.

SLCSE 01
2/12/87

Figure 3-3. SLCSE Toolset.

__General tools.__ This category includes a text editor, command language editor and mail facility, as well as a document formatter.

__Verification tools.__ In the verification tools category, a tracing capability will be provided by the Automated Life Cycle Impact Analysis (ALICIA) system and quality analysis will be provided by the Automated Measurement System (AMS). Other capabilities will be provided by the Ada Test and Verification System (ATVS) for Ada; J73 Automated Verification System (J73AVS) for JOVIAL; RXVP80 for FORTRAN; and Cobol Automated Verification System (CAVS) for COBOL.

__Requirements analysis tools.__ All capabilities in the requirements category will be provided by a new tool to be built for the SLCSE.

__Design tools.__ All capabilities in the design category will be provided by the Software Design and Documentation Language (SDDL) tool.

__Coding tools.__ In the coding category, syntax-directed editing capabilities will be provided by DEC's Language Sensitive Editor (LSE). Other DEC tools will provide compiling, assembling, linking and debugging support, except that the JOVIAL compiler will be provided by Proprietary Software Systems (PSS). Executable assertion translation for Ada will be provided by ATVS; Executable translation for JOVIAL will be provided by J73AVS.

TOOL	SOURCE	Included in Build 1	2	3	4	5	6	7	8
Testing Tools									
Instrumentor									
Ada	Part of ATVS							■	■
JOVIAL J73	Part of J73AVS						■	■	■
FORTRAN	Part of RXVP80				■	■	■	■	■
COBOL	Part of CAVS						■	■	■
Post Execution Analyzer									
Ada	Part of ATVS							■	■
JOVIAL J73	Part of J73AVS						■	■	■
FORTRAN	Part of RXVP80				■	■	■	■	■
COBOL	Part of CAVS						■	■	■
Test Summary Reporter									
Ada	Part of ATVS							■	■
JOVIAL J73	Part of J73AVS						■	■	■
FORTRAN	Part of RXVP80				■	■	■	■	■
COBOL	Part of CAVS						■	■	■
Test Manager	Unnamed - new tool						■	■	■
Configuration Management Tools									
Software Configuration Manager	Unnamed - new tool				■	■	■	■	■
Documentation Manager	Unnamed - new tool						■	■	■
Test Data/Results Manager	Unnamed - new tool						■	■	■
Change Impact Analyzer	Part of ALICIA							■	■
Project Management Tools									
Project Planner	Part of APMS				■	■	■	■	■
Project Tracker/Reporter	Part of APMS				■	■	■	■	■
Software Problem Report Processor	Part of APMS				■	■	■	■	■
Engineering Change Request Processor	Part of APMS				■	■	■	■	■
Environment Management Tools									
Method Script Editor	Unnamed - new tool					■	■	■	■
Menu Editor	Part of WINNIE	■	■	■	■	■	■	■	■
Keypad Editor	Unnamed - new tool			■	■	■	■	■	■
Command Procedure Editor	Unnamed - new tool			■	■	■	■	■	■
Global Command Setup	Unnamed - new tool			■	■	■	■	■	■
Tool Installer/Deleter	Unnamed - new tool						■	■	■
Related Effort Tools									
ATVS								■	■
APMS					■	■	■	■	■
AMS								■	■
ALICIA								■	■
Prototyping Tools									
WINNIE		■	■	■	■	■	■	■	■

NOTE: New tools will be developed under the SLCSE contract; partial shading means prototype version of tool.

Figure 3-4. SLCSE Toolset (Continued).

Testing tools. A Test Manager and Test Data/Results Manager tool will be built for the SLCSE. Other capabilities will be provided by ATVS, J73AVS, RXVP80 and CAVS.

Configuration management tools. A Software Configuration Manager and Documentation Manager will be built for the SLCSE. Change impact analysis capabilities will be provided by ALICIA.

Project management tools. All project management capabilities will be provided by the Automated Project Management System (APMS).

Environment management tools. All environment management capabilities will be provided by new tools built for the SLCSE.

Prototyping tools. Prototyping and special purpose tools will be included in this category, including a window prototyper (WINNIE).

4 SLCSE MAJOR DESIGN CONCEPTS

4.1 User Interface Design Concepts

Data Driven Interfaces. The SLCSE user interface will make substantial use of a sophisticated window management package (WINNIE) and an associated table driver mechanism (MOO). This existing software will make it much easier to design and tailor an easy-to-use interface. The WINNIE and MOO packages support a rich repertoire of windowing operations and do so in the form of input data, not code. See Figure 4-1. This means that the SLCSE design process can easily include a number of successively improved user interface prototypes. This is important because experience has shown that easy-to-use is a subjective sort of requirement; users need some framework to work with in order to provide feedback on user-friendliness.

```
! WINNIE File for SLCSE Command Executive

WINDOW 5 15 47 18 5
 INVISIBLE
 FRAME WITHOUT ID
 FRAME VIDEO "BOLD"
 VERTICAL SCROLL
 TEXT 1 6 " HIGH PRIORITY TOOLS " "REVERSE"
 FORM 1
  ACCESS BY NUMBER WITH RETURN
  HELP 151 2 " Help Information will appear here."

  FIELD 1 3 5 26 "PLAIN" "BOLD REVERSE" PROTECT
   DEFAULT "Project Management (APMS) "
   LABEL 3 1 " 1. " "PLAIN" "BOLD REVERSE"

  FIELD 2 4 5 26 "PLAIN" "BOLD REVERSE" PROTECT
   DEFAULT "Spreadsheet (FCALC) "
   LABEL 4 1 " 2. " "PLAIN" "BOLD REVERSE"

  FIELD 3 5 5 26 "PLAIN" "BOLD REVERSE" PROTECT
   DEFAULT "Text Editor (EDT) "
   LABEL 5 1 " 3. " "PLAIN" "BOLD REVERSE"
```

Explanation of first block: Window #5 is invisible; has a bolded frame; is vertically scrollable and has the words HIGH PRIORITY TOOLS at the top in reverse video. There is one form in the window and the associated help window is #151.

```
! MOO File for SLCSE Command Executive.

IF WINDOW - 1 AND FIELD - 1 THEN UPON
  CRET-VIS 99 166, GO TO 4.

! Window 5 is Project Manager Tools Window.
IF WINDOW - 5 AND FIELD - 1 2 3 4 5 THEN UPON TAB-
  INV 197, VIS 166, RETURN; CRET-INV 197, ADV 164.
IF WINDOW - 5 AND FIELD - 6 THEN UPON TAB-INV 197,
  VIS 166, RETURN; CRET-ADV 30, INV 197, VIS 3.

! Window 6 is Project Manager Tools Window.
IF WINDOW - 6 AND FIELD - 1 2 3 4 THEN UPON TAB-
  INV 197, VIS 166, RETURN; CRET-INV 197, ADV 164.
IF WINDOW - 6 AND FIELD - 5 THEN UPON TAB-INV 197,
  VIS 166, RETURN; CRET-ADV 30, INV 197, VIS 3.

! Window 14 is the exit confirmation window.
IF WINDOW - 14 AND FIELD - 1 THEN UPON CRET-
  CASE OF (TEXT),
    CASE ("Y"), CODE EXIT,
    CASE ELSE, RETURN,
  END CASE.
```

Explanation of first block: If control is at window #1 and field #1 (i.e. the 1st input field), then if the user hits a carriage return, visibilize window #99 and #166 and turn control over to window #4.

Figure 4-1. WINNIE/MOO Data Excerpts.

Environment Instantiation Methodology. Environments, like applications software, have a life cycle. For SLCSE environments, this life cycle is illustrated by the diagrams in Figures 4-2 through 4-5.

First, the SLCSE must be installed. The SLCSE will be delivered to a new installation site with a single working environment - for installation purposes - and several Environment Specifications Packages - for use as default mechanisms in subsequent instantiation activities. SLCSE Installation Personnel will decide whether or not to create additional Environment Specifications Packages beyond those supplied for small, medium and large projects, and will instantiate the first environment for actual project use.

Creating an Environment Specifications Package (ESP) or a Project Environment (PE) involves specifying and/or entering data that describes all the characteristics of an environment: the operating environment (hardware/software configuration), the users, the embedded methods that will govern the environment and miscellaneous information (help, etcetera). ESPs reduce the amount of work involved in instantiating a PE, because they provide reasonable defaults for most data items.

After a PE has been instantiated, it may then be put to use supporting software development activities. It is important to recognize that the SLCSE Database is really two databases:

(1) A Project Database that is a storehouse for information on the software development project

(2) An Infrastructure Database that keeps track of parameters and options associated with the Environment framework

Some users, of course, will deal almost exclusively with ordinary data files. The Programmer role, for example, will involve the ad-hoc creation of source code files. All such files will be known to the Infrastructure Database - because they must show up as valid menu options - but these files will be unknown to the Project Database until they are officially "imported" into the appropriate data structures of that database.

There will eventually come a time when a PE/ESP must be updated. This will involve the same sorts of activities as in instantiation except that users must be locked out of the PE/ESP while the update is effected.

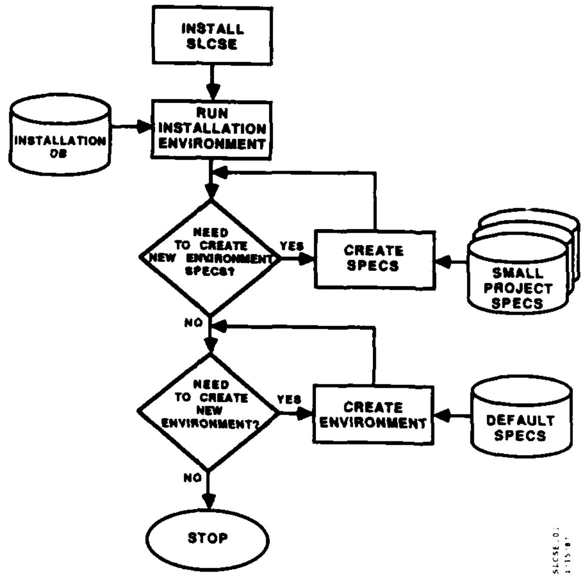


Figure 4-2. Install SLCSE Thread.

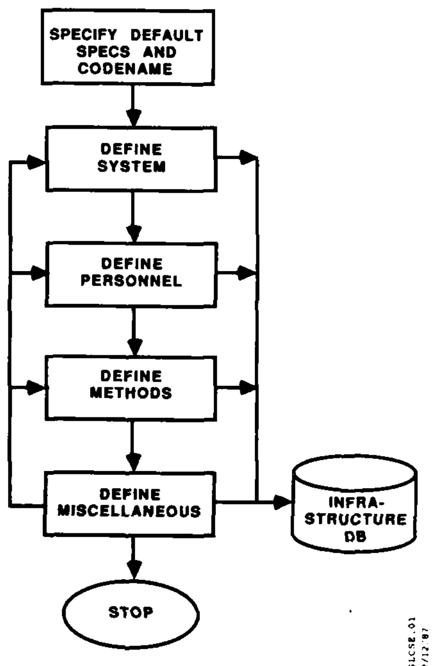


Figure 4-3. Create Specs/Environment Thread.

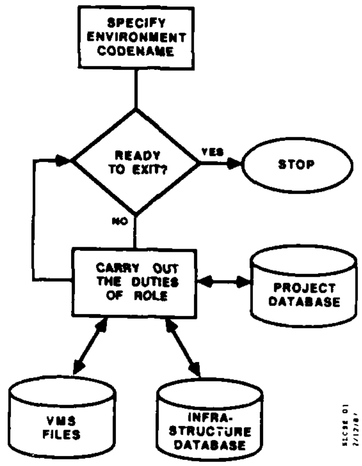


Figure 4-4. Use Environment Thread.

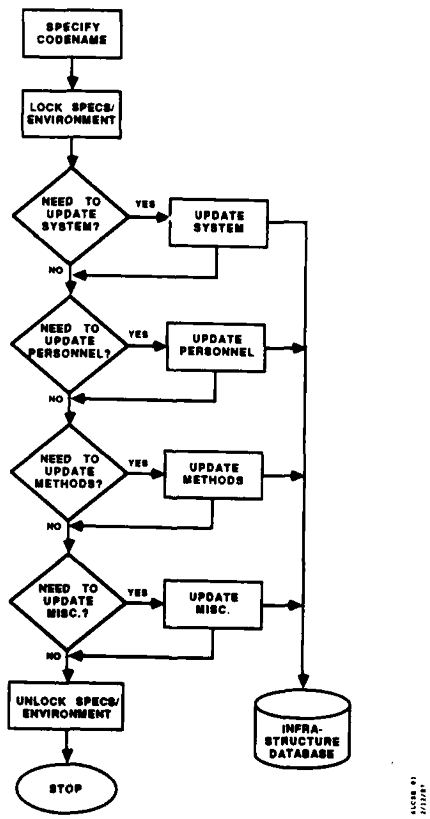


Figure 4-5. Update Specs/Environment Thread.

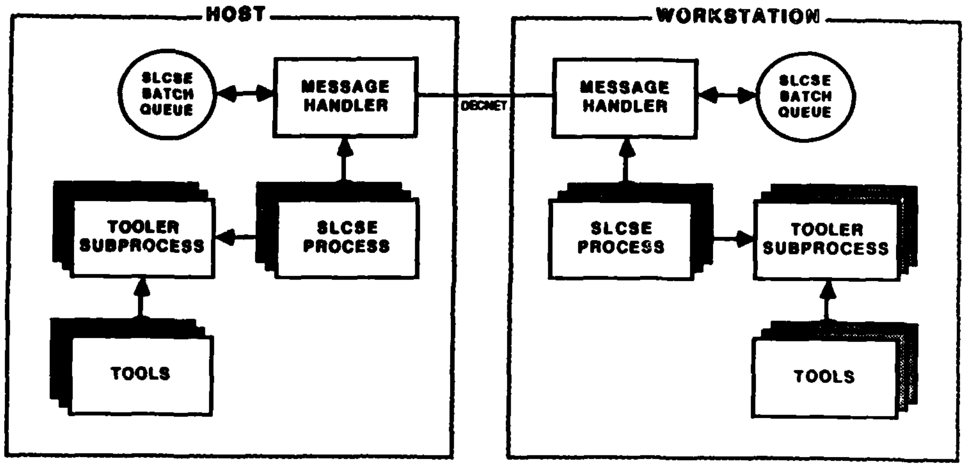


Figure 4-6. Scheduler Mechanism.

Scheduler Mechanism. The Scheduler Mechanism has been designed in two parts: a Tooler and a Message Handler. Refer to Figure 4-6.

Each SLCSE process, during its initialization, will spawn a Tooler subprocess that will communicate with the main process via mailbox services. The Tooler subprocess will hibernate until sent a command from the main process. It will execute this command, taking temporary control of the screen. This is the manner in which standalone, executable images will be controlled from SLCSE processes.

Also, each computer in the environment will have a single, detached Message Handler process. All SLCSE processes will establish communication with this Message Handler, which will take care of "no wait" requests via a SLCSE batch queue and will handle the distributed aspects of an environment via network utilities.

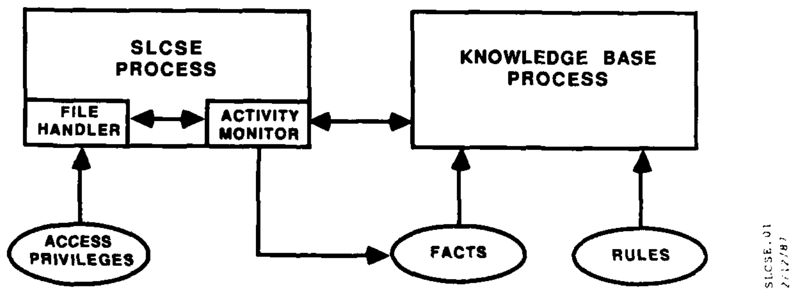


Figure 4-7. Methods Mechanism.

Methods Mechanism. The Methods Mechanism has been designed in three parts: an Activity Monitor, a File Handler and a Knowledge Base. Refer to Figure 4-7.

The Activity Monitor keeps track of user requests and sequences of user requests, deciding when to invoke the File Handler and Knowledge Base and when to update information about the current situation (i.e. the facts). The File Handler uses operating system utilities to prevent unauthorized access to files. The Knowledge Base - implemented in Prolog - will store and enable assessment of procedural type rules governing an environment. This will be the mechanism whereby a project manager can both encourage and enforce adherence to project specific methods.

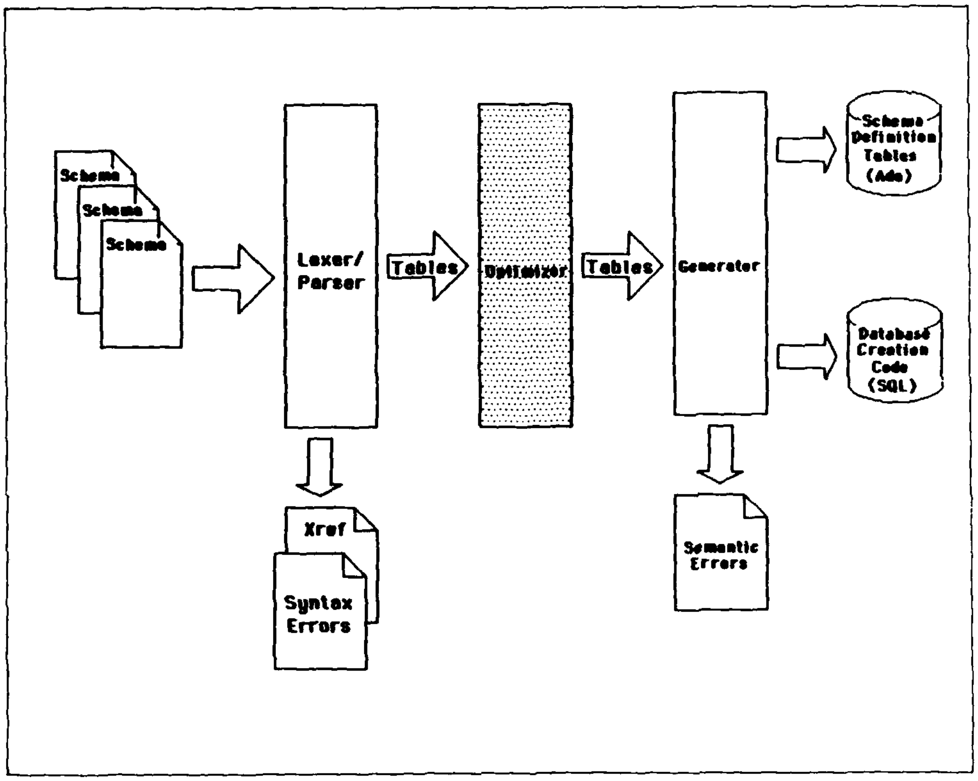


Figure 4-8. Schema Language Compiler.

4.2 Database Design Concepts

SMARTSTAR/OMNIBASE. There is only one commercial database management system that uses the Britton Lee Database Machine: OMNIBASE. Fortunately OMNIBASE has a lookalike - SMARTSTAR - that can make use of ordinary disk units. The SLCSE will be developed using SMARTSTAR, but will be installed at the RADC Computer Facility using OMNIBASE.

Schema Generator. The SLCSE Database Schemas and Subschemas will be developed in the form of Entity-Relationship (ER) models which will be expressed in an Ada-like language. This language will be compiled as illustrated in Figure 4-8.

The Lexer/Parser will transform the input text into tokens, parsing them into internal tables of information. In the process, it will report syntax errors and produce a cross-reference listing of entities, attributes and relationships, showing where they are defined and used.

The Optimizer will select attributes to be indexed and will select representations of relationships, so as to optimize runtime access speeds.

The Generator will read the tables produced by the Lexer/Parser or the Optimizer and generate the final outputs of the compiler:

(1) Database Creation Code, a file of Structured Query Language (SQL) statements that will create and initialize an empty database that matches the input schema(s)

(2) Schema Definition Tables, Ada packages that will be used for type checking by SLCSE applications

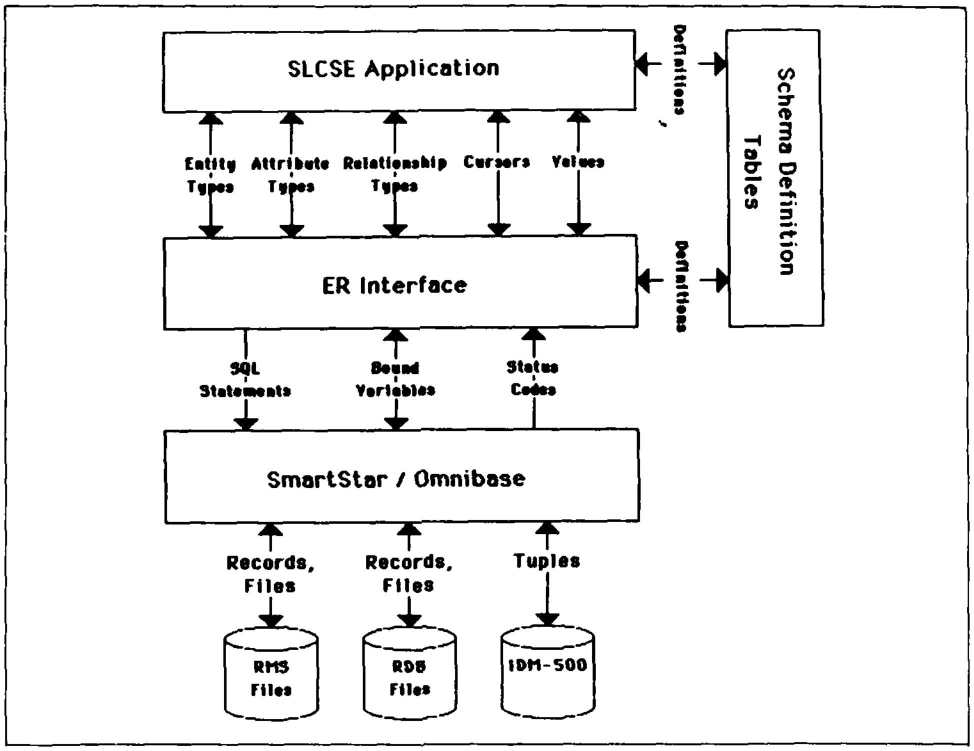


Figure 4-9. Database Interfaces.

ER Interface. Ultimately, a SLCSE application will want to store and retrieve information from the SMARTSTAR/OMNIBASE Database, still making use of the ER model forms. Using the Schema Definition Tables produced by the Schema Language Compiler, a SLCSE application will communicate an ER-type request to the ER Interface, which will in turn communicate the request to the underlying database, SMARTSTAR/OMNIBASE. See Figure 4-9.

4.3 Toolset Design Concepts

Tool Integration Guidelines. Guidelines for coupling tools with the SLCSE will be published as part of the Software Programmer's Manual.

Directly coupled tools will conform to user interface standards set forth in the guidelines and will know about and make direct use of the SLCSE Project Database. Directly coupled tools, then, will look and act like extensions to the SLCSE framework.

Indirectly coupled tools will maintain their original user interface and will neither know about nor make direct use of the SLCSE Project Database. This means that there will be a non-SLCSE look to such tools and that interface mechanisms may have to be designed to preprocess or postprocess data for their internal, non-SLCSE databases.

5 SLCSE IMPLEMENTATION AND RESULTS TO DATE

5.1 The Eight Incremental Builds

As previously mentioned, the SLCSE is being developed in a series of eight incremental builds spanning a twenty-four month period. Refer to Figure 5-1. Builds 1 through 7 are three months in duration (a quarter system). The last developmental quarter will be a one month build (Build 8) followed by two months of installation activities. An incremental build approach was chosen to enable early prototyping of critical functions and to provide an "early warning" mechanism for an ambitious development schedule. New capabilities are demonstrated at the close of each quarter; if the required capabilities cannot be demonstrated, the project is behind schedule!

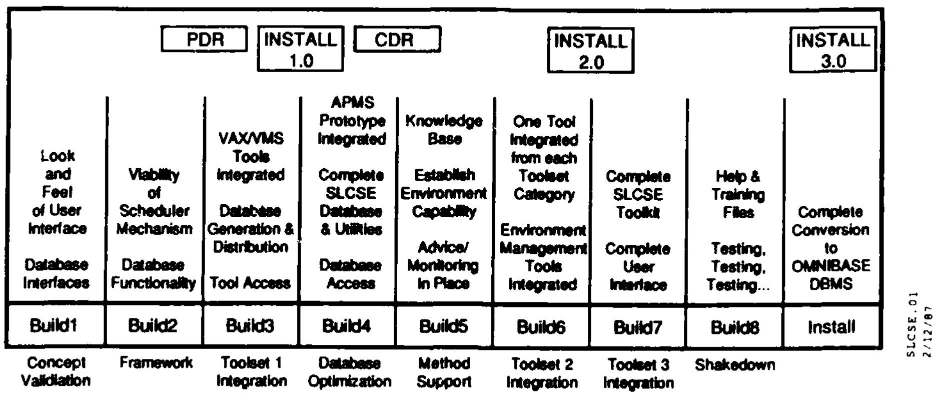


Figure 5-1. SLCSE Builds and Major Milestones.

Build 1 - Concept Validation - will demonstrate the "look" and "feel" of the User Interface and establish specifications for the various Database Interfaces. Build 2 - Framework - will prove the viability of the Scheduler mechanism and will demonstrate limited Database functionality. Build 3 - Toolset 1 Integration - will integrate standard operating system tools into the SLCSE, will demonstrate full Database functionality and will introduce a Tool Access mechanism (for indirectly coupled tools). Build 4 - Database Optimization - will complete the SLCSE Database Subsystem and associated database management utilities, will demonstrate a Database Access mechanism (also for indirectly coupled tools) and will integrate a prototype Automated Project Management System (APMS). Build 5 - Method Support - will implement a SLCSE Rule Base supporting methodology-based advice and monitoring features and will demonstrate the capability to instantiate a new environment. Build 6 - Toolset 2 Integration - will integrate all Environment Management tools and at least one tool from each of the other tool categories. Build 7 - Toolset 3 Integration - will complete the SLCSE User Interface and will integrate the complete SLCSE Toolkit, including tools being built under other Air-Force-sponsored contracts. Build 8 - Shakedown - will install help and training facilities for the SLCSE and will include production testing of the system.

The Preliminary Design Review (PDR) and Critical Design Review (CDR) will be held at six and twelve months into the project, respectively. However, because of the incremental builds, both of these reviews will involve preliminary designs, detailed designs and actual code.

Three versions of the SLCSE will be installed at the Rome Air Development Center computer facility. Version 1.0, at ten months into the project, will be geared toward the support of coding functions. Version 2.0, at nineteen months into the project, will support coding, project management and SLCSE installation functions and will demonstrate some functionality across the life cycle for at least one of the supported programming languages. Version 3.0, at twenty-three months into the project, will support the complete software development life cycle for Ada, JOVIAL, FORTRAN and COBOL.

5.2 Build 1 Results

Build 1 was completed in early December, 1986, as scheduled.

A SLCSE user interface was designed and demonstrated. Referring to the sample display in Figure 5-2, the User Interface makes full use of advanced windowing technologies. The current process (either SLCSE or one of the SLCSE tools) is indicated at the top of the screen, along with the current user role (on the far right) and the current object (on the far left), if any has been specified. The bottom of the screen is used consistently for prompt, error and comment messages. The SLCSE may be exercised in either Menu Mode or Keyword Input Mode. Every Menu option has an identical Keyword option. However, in Keyword Input Mode, the user only has to enter as many characters as are needed to uniquely identify the option. In Menu Mode, command buttons appear underneath the top line with the following meanings:

OBJECTS -	for a list of available files/database items
TOOLS -	for a list of available software development tools
PROCEDURES -	for a list of available software development procedures
ADVICE -	for various kinds of advice on software tasks
MODE -	to change user modes/parameters
HELP -	for detailed help on how to use the SLCSE
EXIT -	to get out of the SLCSE

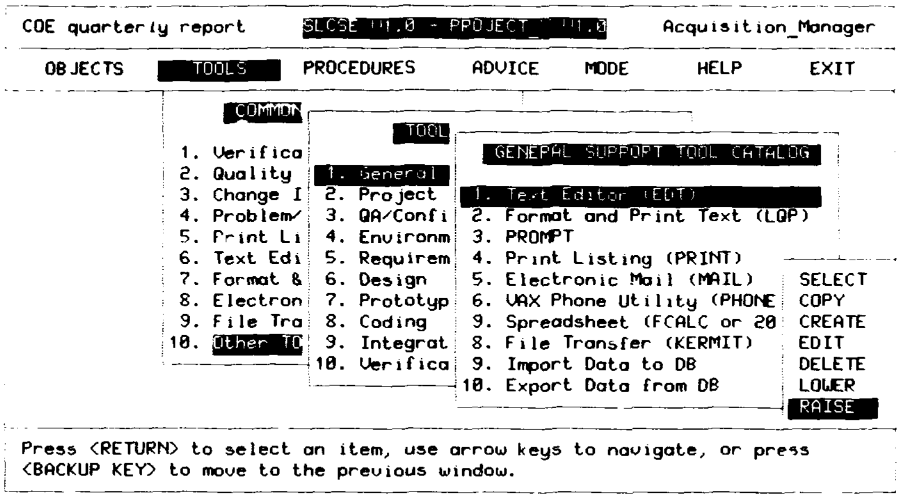


Figure 5-2. Sample SLCSE Display.

Windows drop down from the command button line. For objects, tools and procedures - referred to collectively as components - the windows cascade down and to the right in three tiers. The highest level menu displays the components commonly used on a day-to-day basis; the next level menu displays a list of component categories; the lowest level menu displays a catalog of all components available to the user in a particular component category. These displays are tailored for each user role so that, for example, what a Programmer sees is different than what a Project Manager sees. Additionally, each user may tailor their own particular displays by raising components from the component catalog to the "commonly used" menu or lowering components from the "commonly used" menu to the component catalog. While many of the components available to the user are under database control, users may also work with private, non-controlled components. These may be created, copied and deleted by bringing up a special menu as shown in the lower righthand corner of Figure 5-2.

The tailoring capabilities supported by the SLCSE User Interface will go a long way toward fostering user acceptance of SLCSE environments.

6 SLCSE TESTING & VERIFICATION

Development of the SLCSE will advance the state-of-the-art in software tool integration technology and software development life cycle support by undertaking the design and implementation of highly sophisticated and complex Man-Machine-Interface and database components as well as the difficult integration of new software development tools with existing off-the-shelf capabilities. There is moderate technical risk in developing an environment such as the SLCSE, not only due to the required functionality, but also due to performance concerns; a sluggish environment, no matter how effective and easy-to-use, will not be accepted by users. With this element of risk, testing and verification activities to be performed by both the contractor and the Government are critical.

Verification is the iterative process of determining whether the developing product, at each phase of the life cycle, fulfills the requirements established by the previous phase. The process provides a level of assurance that serious errors do not exist, that all user requirements have been supported, and that critical functions of the final product will not fail during operational use. The incremental build approach for developing SLCSE will provide a series of interim products which will give both the contractor and the Government several test and verification opportunities for evaluating the adequacy of SLCSE design concepts, implemented functionality, and performance. These opportunities also make possible timely corrective actions, as necessary. Each of the eight SLCSE builds will be reviewed by the Government. Versions 1.0 and 2.0 of SLCSE, which will consist of "threads" or "vertical slices" of functionality, will be formally delivered to the Government and acceptance-tested. The final delivery, SLCSE Version 3.0, will be tested in accordance with a Government-approved Software Test Plan, Software Test Description, and Software Test Procedures, each of which will be prepared in accordance with DoD-STD-2167.

Proof of SLCSE's sufficient functionality and system responsiveness will be initially demonstrated by its ability to support itself; as development of SLCSE proceeds, available components of SLCSE will be used by the contractor to implement

subsequent SLCSE components. As a result, upon delivery of SLCSE Version 3.0, a significant subset of SLCSE will exist and be supportable within itself.

7 SLCSE FUTURE DIRECTIONS

Beyond the scheduled SLCSE 3.0 delivery, the SLCSE will continue to evolve. The SLCSE toolset will be considerably expanded and the supporting technologies, especially for the Project Database, will continue to be developed. The SLCSE will also provide an important R&D base for state-of-the-art advances in life cycle software development technology. The key to future advances and the potential for realization of advanced operational life cycle software engineering capabilities lies with the project database. By providing for the efficient storage, management, and retrieval of all data generated throughout the life cycle, the database makes possible the development of several high-payoff capabilities. Such technologies as life cycle impact analysis and assessment, automated documentation generation and data collection for quality and productivity measurement will be the focus of much of the future research.

Impact analysis. The process of impact analysis and assessment is required whenever a change is contemplated to an MCCR system. Historically, the extent of changes to an initial MCCR system has ranged from adding minor requirements to changing the entire system concept. Changes can come about because of new or updated requirements, unexpected results, error detection/correction, inter- and intra-system interface anomalies, and performance characteristics that are not responsive to mission-derived specifications. Such changes can significantly contribute to cost and schedule overruns. Effective change impact analysis during development and post-deployment support can help overcome these difficulties by providing support for (1) adapting to changing requirements, (2) evaluating proposed design changes or alternative designs, (3) correcting errors, and (4) enhancing MCCR systems. Automated impact analysis and assessment will improve the resulting quality of MCCR software by reflecting changes more completely and reliably, without adding undesirable side effects. While SLCSE 3.0 will have a limited impact analysis capability supported by a small set of deterministic algorithms, a more advanced capability utilizing heuristic and stochastic algorithms is planned.

Automated documentation. An MCCR system development, requires the generation of documents which describe the software product and which conform to military standards such as DoD-STD-2167. Documentation accounts for 40% of total software development costs, but the lead time required for documentation updates hampers the timely transition of a system update to the field. Thus, it is important that advanced environments such as SLCSE provide documentation support. The SLCSE project database will be capable of storing all pertinent information generated during the software life cycle. Documentation generators are planned for each of the DoD-STD-2167 Data Item Descriptions (DIDs). These will be available to extract information from the project database (user-generated text, graphics, tool-generated tables, etc.) and produce or update the relevant documents.

Quality and productivity measures. The measurement and assessment of product quality and development productivity is becoming an important concept in the control of a software development project. The current state-of-the-art is limited to the existence of a few tools which provide both the data collection and measurement/assessment functions. The data is normally collected, if not manually, through the use of static analyzers and post-processors which filter appropriate data from an individual software tool's data files. Future research will be focused on the automated collection of pertinent data, throughout the entire software life cycle, in a manner which is as transparent as possible to the users of environment tools. This approach not only supports the complete collection of correct and consistent data, but also relieves measurement and assessment tools of the responsibility and burden resulting from the storage, management, and retrieval of the data.

8 CONCLUDING REMARKS

The SLCSE is but one project in a long line of successful (and unsuccessful) efforts aimed at establishing software development environments. It is, however, unique in many respects:

Distributed. The SLCSE environment operates from a host computer and any number of workstations.

Multi-lingual. The SLCSE will support software development in any (or all) of four languages - JOVIAL, Ada, FORTRAN and COBOL.

Embedded methods. The SLCSE includes a spectrum of capabilities involving methods, everything from advice on how to go about doing a task to monitoring user actions to ensure adherence to particular project rules.

Detailed, high performance database. The SLCSE Project Database includes the complete information content of all of the project documents described in DoD-STD-2167 data item descriptions (DIDs), implemented on a high performance IDM-500 Database Machine.

ER front end. Access to SLCSE data is via an English-like, easy-to-understand model, the Entity-Relationship (ER) model. This makes use and maintenance of the detailed database more straight forward.

Evolving toolset. The SLCSE has been designed to easily accommodate new and existing tools. This makes it a perfect starting point for developing unique MCCR environments.

Life cycle software development environments are one of the most important areas of research in software engineering today. They are a "precursor" technology, a technology that must come to fruition if certain other technologies - like automated simulation synthesis - are to succeed. Unfortunately, even though the goal is clear, the path toward achieving that goal is treacherous, littered here and there with abandoned shells and crumbling monoliths. Those who have succeeded in some measure are those who have left the path altogether to create environments with limited scope or functionality.

SLCSE is not among these efforts. Its goal of a multi-lingual, integrated software development environment across the complete life cycle is as ambitious, if not more so, than those of previous efforts: Yet SLCSE is different. It preserves the goal, but chooses to achieve that goal in small increments, finding solutions to the tough problems first and maintaining an unprecedented flexibility in design.

The SLCSE project is like the successful Design Engineer who, given a task, would immediately set down the first thoughts that came to him and present them to a colleague. His colleague would, of course, tear this ill-conceived concept to shreds. The Design Engineer would carefully evaluate all this criticism and shortly come up with a revised concept, which he would present to a different colleague for review. This iterative process would continue until the concept was completely thought out. This Design Engineer was given only the most difficult assignments and always succeeded. This person actually exists and was described in recent literature to illustrate the winds of change that are sweeping the engineering disciplines.

The iterative design technique - what we generally refer to as incremental builds and prototyping - is an invaluable aid for projects on the forefront of technology. SLCSE, with its eight incremental builds and intermediate prototypes, has fully embraced this new approach and, after a successful first build, has a foot solidly on the goal-directed path. Only time will tell if the footsteps are sure enough, or the stride long enough, to take us into a new age of software innovation.

DISCUSSION

R.Guoit, FR

What are you doing about the problem of performance?

Author's Reply

We are using a database machine, which helps a little. Also, we are using all that database technology has to offer in terms of prefetching and catching data — anticipating what the user will need next. Finally, every major component of the SLCSE will undergo an optimization pass to ensure the production code is as efficient as possible.

Question

How is the SLCSE related to specific MCCR programs?

Author's Reply

The Rome Air Development Center sponsor is tasked with developing state-of-the-art technologies, which are then provided to operational groups with the Air Force. The SLCSE will be of great interest to those groups developing MCCR software as soon as it has been proven production-worthy.

M.Muenier, FR

Ada is one of the languages supported by the SLCSE. There is an ongoing effort to standardize environment to support Ada, namely the CAIS. What is your position on this point?

Author's Reply

We are very familiar with the work on CAIS and agree with the basic idea; however, we feel that it has a long way to go before it will be of use to applications developers. Current implications of the CAIS are a subset of the CAIS specification, are agonizingly slow, take up a tremendous amount of internal and external memory, and are not truly portable (having small amounts of system-dependent code).

P.Simons, US

Does the SLCSE support the development of embedded code?

Author's Reply

Embedded code very often requires tools such as target simulators and cross-assemblers. Such capabilities are not a part of the basic SLCSE tool set but could, of course, be added in the future.

G.Bouche, GE
Will the SLCSE be available to the NATO community? If so, when?

Author's Reply
Yes, both General Research, the prime contractor, and the Rome Air Development Center, the sponsor, are committed to the expeditious transfer of this technology.

D.Schimsky, US
Do you think that the lack of portability of the tools (e.g., VAX/VMS) is a handicap? What do you think of recent efforts to standardize on an OS called POSIX?

Author's Reply
A great number of military installations are committed to VAX/VMS, so we think that the SLCSE has wide applicability. On the issue of standard OSs, we have heard for the last 10 years that one OS or another would become a "standard," but we have yet to see a real standard emerge due to the vendors' reluctance to support such a move.

M.Kayton, US
For what purpose does your customer intend to use the SLCSE after you deliver your three builds? Your paper refers to development of MCCR software, and you addressed the development of tools.

Author's Reply

(1) First, we must clarify two points. The first point is that the SLCSE is being implemented in a series of eight incremental builds consisting of three formal deliveries. The final delivery from General Research (GRC) will be designated SLCSE Version 3.0. Versions 1.0 and 2.0 will only be used in-house at the Rome Air Development Center to provide effective design feedback to the contractor. It is intended that Version 3.0 will be a releasable product. The second point is that the Rome Air Development Center, a US AFSC R & D laboratory, is GRC's direct customer.

(2) The primary goal of the current SLCSE development effort is to provide the environment "framework" (i.e., user interface, project database, executive, etc.) within which software development tools can be integrated. While the SLCSE will be delivered with a comprehensive tool set, it will by no means be extensive. The SLCSE will be used by the Rome Air Development Center to support, in a technical way, future software engineering and knowledge-based R & D. Because of its state-of-the-art life-cycle project database component, it is an excellent vehicle for advancing technology in areas such as knowledge-based systems, project management, software quality specification, data collection, quality analysis/assessment, automated documentation generation, and change impact analysis.

(3) The Rome Air Development Center also has potential "customers" for SLCSE who are involved in MCCR software development and have a need for life-cycle support. Beyond the current General Research contract, it is very likely that the SLCSE will be enhanced through the integration of necessary tools to support the embedded software development needs of these customers. The first likely addition will be support for MIL-STD-1750A.

DEVELOPMENT OF AN AIRBORNE FACILITY FOR ADVANCED AVIONICS RESEARCH

by
N. van Driel
National Aerospace Laboratory NLR
P.O. Box 90502, 1006 BM Amsterdam,
The Netherlands

ABSTRACT

The present generation of aircraft is equipped with a number of advanced avionics systems like digital Flight Management or Mission Computers, Flight Control Computers and Electronic Flight Instrument Systems (EFIS). In the near future new systems may be introduced as the Microwave Landing System (MLS), Navstar Global Positioning System (GPS) and digital data link systems (SSR-Mode S or via satellites). In order to derive the maximum benefit from these systems new procedures will have to be developed. Extensive simulator and inflight testing is required to determine the optimal use of the new avionics systems and to establish new operational procedures.

The Netherlands National Aerospace Laboratory NLR performs research in the area of MLS using a moving-base simulator and is planning to use its Fairchild Metro II research aircraft for airborne testing. To this purpose a multi-year program has been started called ART (Avionics Research Testbed) in the framework of which the aircraft will be equipped with systems including:
. Programmable Electronic Flight Instrument System
. Programmable Flight Management Computer
. Programmable Flight Control Computer
. Navstar GPS
. MLS
. Digital data link.

The fully equipped aircraft will be suited to perform inflight research in such areas as:
. establishing MLS procedures
. use of Navstar GPS as a navigation and approach system
. conventional and unconventional presentations on the EFIS
. 4-D navigation.
Flight tests with the system developed in the first phase of the ART-project, based on EFIS with standard navigation sensors, are scheduled for July 1987.

LIST OF ABBREVIATIONS

ADF	Automatic Direction Finder
ADI	Attitude Director Indicator
ART	Avionics Research Testbed
ATC	Air Traffic Control
BCP	Bezel Control Panel
CDU	Control Display Unit
DADC	Digital Air Data Computer
DEU	Display Electronics Unit
DME	Distance Measuring Equipment
DU	Display Unit
EFIS	Electronic Flight Instrument System
FCC	Flight Control Computer
FCS	Flight Control System
FMC	Flight Management Computer
FMS	Flight Management System
GP	Glide Path (ILS)
GPS	Global Positioning System
HSI	Horizontal Situation Indicator
IAS	Indicated Air Speed
ICAO	International Civil Aviation Organization
IFR	Instrument Flight Rules
ILS	Instrument Landing System
INS	Inertial Navigation System
IRS	Inertial Reference System
LOC	Localizer (ILS)
MLS	Microwave Landing System
ND	Navigation Display
NLR	Netherlands National Aerospace Laboratory
PFD	Primary Flight Display
PPS	Precise Position Service (GPS)
RA	Radar Altimeter
RCP	Remote Control Panel
RMDU	Remote Multiplexing Digitizer Unit
SPS	Standard Position Service (GPS)
SSR	Secondary Surveillance Radar
VOR	VHF Omnidirectional Radio Range

1. INTRODUCTION

The new generation of military and civil aircraft is equipped with a number of advanced avionics systems as Flight Management and Flight Control Systems and Electronic Flight Instrument Systems (EFIS). In the near future new systems can be expected to be introduced, including the Microwave Landing System (MLS), satellite navigation and communication systems and digital data link systems (SSR-Mode S and satellite data link).

In conjunction with the introduction of these systems new procedures and air traffic control (ATC) concepts will have to be established in order to derive maximum benefit from the use of these systems. For example, in case of MLS new rules and procedures are required for the definition of safe and fuel efficient approach paths (even during equipment malfunctions), for the selection by the pilot of his reference approach path, for the way in which he receives and uses guidance information, and for ATC for the selection of different approach paths within the MLS approach sector for the mix of different categories of aircraft.

The use of EFIS is considered to be an important prerequisite for the pilot to fly a complex MLS approach. It may be anticipated that, especially for curved or segmented approaches, presentation formats will be required quite different from those presently seen on the flight deck. An important consideration in these developments is that the pilot is provided with safe and unambiguous information with regard to his reference track even during (MLS) equipment malfunctions.

The navigation and performance management potential of Flight Management Computers (FMC) exceeds the capabilities of present-day ATC systems, as a result of which very often non-optimal profiles are assigned. In addition, the increasingly accurate navigation capabilities achievable today, especially with the advent of satellite navigation systems, call for a more flexible route structure based on the available area navigation (RNAV) capabilities.

What is required is a new ATC-aircraft relationship with the aircraft indicating to the ATC system their preferred horizontal (RNAV) and vertical flight paths, based on an individual flight profile optimization in the FMC, and ATC using this information to assign a flight profile to a particular aircraft based on a collective optimisation of the total traffic situation. A digital data link for communication between the aircraft (FMC) and the ground (ATC computer) is an essential prerequisite in this concept.

Although extensive research has already taken place, a considerable amount of testing is still required to support the realization of concepts and the definition of procedures in the areas indicated here.

Over the last decade the Netherlands National Aerospace Laboratory NLR has been involved with research in many of the areas indicated above. For example, extensive flight simulator research has been carried out to support the definition of MLS approach procedures (Ref. 1). In order to expand the capabilities for research in the areas discussed above, work is currently under way at NLR to realize, besides the moving base research flight simulator, an airborne avionics research facility and an ATC research simulator (figure 1). Coordinated operation is aimed for by carefully attuning the capabilities of these three research facilities.

For the development of the airborne facility the Avionics Research Testbed (ART) project has been created, which is the topic of this paper.

2. ART SYSTEM CONCEPT

NLR operates two research airplanes: a Beechcraft Queen-Air and a Fairchild Metro II. For a number of reasons (size, payload, operational flexibility) it was decided to use the Metro airplane (figure 2) for the ART-project.

The Metro II is a twin-engine turboprop airplane with a gross weight of 14000 lbs, operating in the speed range between 120 and 250 kts. The airplane has been modified extensively for aerospace research purposes.

The goal of the ART-project is to equip the Metro with a number of advanced avionics systems in such a configuration that it will be possible to perform operational research on the use of these systems.

The main areas for application of the ART-system include:

- research on MLS approaches (e.g. interception angles, segment lengths and orientations, equipment requirements, etc.),
- research on the use of EFIS, both for conventional and unconventional presentations,
- research on the use of digital data links,
- research on the use of FMC's,
- research on the use of satellite navigation systems (e.g. Navstar GPS) especially for RNAV operations and approaches.

In order to be able to perform research in these different areas a flexible avionics system is required of which both hardware and software can easily be adapted to the requirements of the particular test objectives. This means that NLR must have access to and full control over the software and that software modifications can be made by NLR. The consequences of this requirement are discussed in the next section.

A block diagram of the basic system is depicted in figure 3. A number of navigation systems, together with an attitude reference system and air data system, provide information to a programmable FMC. Based on this information the FMC determines the aircraft position and compares it to the desired position or desired track derived from the flight plan. Differences in position, altitude, velocity and, in case of 4-D navigation, time, are then used by the programmable Flight Control Computer (FCC) to generate steering and power setting commands based on control laws for the airplane. Steering commands are presented to the pilot on the Primary Flight Display (PFD) of the programmable EFIS, while coupling to the autopilot may be implemented at a later stage of the program. The navigation information of the FMC is pictorially displayed on the second EFIS display, called the Navigation Display. A digital data link system is integrated with the FMC while a digital data recording system is installed for post flight analysis of test data.

The ART system is a research tool rather than a standard aircraft avionics package. As the Metro airplane is used also for other research projects, it was a requirement from the beginning that installation of the ART system would not comprise the IFR capability of the airplane. For that reason interaction between the ART system and standard avionics systems has been limited to the maximum extent possible. This consideration also resulted in the decision to redesign only the right hand side of the cockpit instrument panel and leave the left hand side unchanged.

3. MAJOR SUBSYSTEMS

The research concentrates on the use of MLS, GPS, digital data link, FMC/FCC and EFIS. These systems will briefly be discussed in the following chapters.

3.1 MLS

The purpose of MLS is to provide an approach and landing system with improved performance compared to the present Instrument Landing System (ILS). Worldwide replacement of ILS by MLS is scheduled for 1998 (Ref. 2). The major advantages of MLS are that it does not have the siting problems of ILS, that it is less vulnerable to interference and that it provides the capability for using approach paths different from the single straight ILS localizer/glideslope approach path. The MLS provides accurate proportional guidance in an azimuth sector covering at a maximum 40 degrees on both sides of the extended runway centerline and between 0.9 and 7.5 degrees in elevation (Ref. 3). Table 1 provides some information on the advantages of MLS over ILS.

The aircraft position relative to the runway can be derived from the MLS azimuth and elevation information together with range information from a DME or precision DME/P station or, possibly, from GPS derived positions.

MLS includes a ground-to-air digital data link for the transmission of data to the aircraft. There are two categories of data, basic and auxiliary. The basic data category includes information regarding azimuth and elevation coverage supporting simple straight-in approaches or approaches in which the pilot selects the approach azimuth course and simple missed approach procedures. So far, the auxiliary data category includes information on offsets in azimuth and elevation antenna position and in DME position, to support area navigation (RNAV) operations. However, additional auxiliary data word capacity is available. Some additional data words will be used in an upcoming application of the ART system directed towards research on a procedural MLS interception procedure.

In the Netherlands the civil aviation authorities are planning to install MLS ground equipment for one runway at Schiphol Amsterdam airport in 1989 to obtain experience with the system. NLR is at the moment exploring the market for airborne MLS equipment with the aim to integrate MLS into the ART-system in 1988.

3.2 Navstar GPS

The Navstar Global Positioning System (GPS) is a satellite navigation system that provides highly accurate position, velocity and time information to suitably equipped users at all times and at any place on or near the earth. It is a passive system which means that users just receive information and do not have to transmit. The system has been developed by the United States Department of Defense, which department also exercises control over the access to the system. Only authorized users will have access to the full accuracy of GPS. Table 2 gives an overview of the different accuracies available (Ref. 4).

Prototype GPS satellites have been launched since 1978 for testing purposes. Navstar GPS is currently scheduled to be operational by 1991 with 18 operational satellites in orbit.

The present constellation of prototype (Block I) GPS satellites provides ample opportunity to use GPS as an accurate position reference system for a number of projects and to test its usefulness for such applications as low level operations and as an approach system. NLR has procured a civilian (C/A-code) single frequency sequential receiver and has obtained good positional accuracies on a number of flights over the Netherlands.

It must be noted that for aviation use, be it military or civilian, the most important benefits of GPS are derived from integration with other navigation systems, especially an Inertial Navigation System (INS). Research with the ART system will therefore especially be directed towards the use of GPS as part of an integrated system

3.3 SSR Mode S

The Secondary Surveillance Radar System (SSR) as in use today has, because of its transmission of identification and altitude data, provided a considerable increase in the safety of aviation and in the capacity of the ATC-system. The carriage of SSR transponders with 4096 codes and altitude encoding capabilities has become mandatory for IFR controlled flights, both military and civilian, in many areas around the world.

However, the increase in traffic has resulted in difficulties with the operation of SSR in busy areas (interference, fruiting, garbling, overinterrogation, airplanes switching codes, etc.). Although modern techniques may solve a number of these problems, some of the difficulties are inherent to the present SSR system.

Two methods for SSR enhancement have been approved by the International Civil Aviation Organization (ICAO) to satisfy future requirements: the first one, monopulse, serves to improve azimuthal accuracy, the second method adds a selective address and data link capability (Mode S) (Ref. 5). Besides for solving the difficulties encountered with the present SSR system the data link capability could prove to be an essential component in the future ATC system as discussed before. Research on how to use this data link as a communication medium between the FMC and ATC is required and is one of the potential applications of the ART system.

3.4 Flight Management Computer

One of the tasks of the programmable Flight Management Computer (FMC) in the ART-system is to provide a flexible tool for performing research on:

. integration of new sensor systems (MLS, GPS, SSR Mode S),
. pilot-FMC communication,
. ATC-aircraft integration/communication,
. EFIS preprocessing, especially for unconventional 3-dimensional presentations.

The fact that this FMC is used for research implicates that software should be easily accessible and changeable. For this reason a programmable general purpose airborne computer is used rather than a commercial FMC. The emphasis for this computer is on navigation management rather than performance management, as the latter function is not directly within the scope of this project. Besides navigation management the computer also incorporates a number of FCC functions.

It was decided to use a ROLM 1664 computer for this task as NLR has obtained considerable experience with the development of hardware (interfaces) and software for this computer in the on-board data recording system developed for the Fokker 50/100 certification flight test programs. The ROLM 1664 is a 16-bit processor with a 64 K words core memory and hardware floating point processor.

3.5 EFIS

The programmable EFIS used in the ART system is developed by Sperry and consists of two Display Units (DU) with integral Bezel Control Panel (BCP), a Remote Control Panel (RCP) and a Display Electronics Unit (DEU). A block diagram is shown in figure 4.

The hybrid (stroke/raster) color Display Unit has an active area of 6.5 x 6.5 inch and presents formats based on data from a number of aircraft systems that is processed through the DEU. The DEU receives sensor data through ARINC 429 serial data inputs, ARINC 708 weather radar data through a 1 MHz serial data input, and source selection and program data through discrete inputs. The BCP, containing 16 pushbutton switches plus 2 potentiometers on each DU, and the RCP provide great flexibility in the selection of different display modes or parameters during the flight.

The formats to be displayed are controlled by software in the DEU. Graphics and raster data is transmitted by the DEU to the DU, via a 1 MHz high speed serial bus. A single DEU can simultaneously provide primary flight information to one DU and navigation information to the other DU. Figure 5 shows both DU's with a representative Primary Flight Display and Navigation Display format.

The unique capability of this EFIS is that the display format is in-house programmable. The system provides the flexibility to define the complete composition of a dynamic display in terms of:

. parameters,
. symbols and characters,
. symbol location,
. parameter priorities,
. colors.

New application software can be developed by NLR personnel on a Software Development Station after which it is downloaded to the DEU through an IEEE-bus. The hardware of this station consists of an IBM AT2 PC with co-processor, 20 Mbyte hard disk and 1.2 Mbyte floppy drive, a monochrome and a color display, printer and tape streamer. The software environment consists of a number of programs and utilities mainly provided by Sperry. Figure 6 shows some of the hardware of this Software Development Station.

4. PROJECT PHASES

Due to the complexity of the project, the variety in project goals and the differences in system components the ART project has been divided in a number of relatively independent phases. The following project phases are presently foreseen:

1. Installation and operation of EFIS, with one display unit for display of primary flight information, and supporting systems,
2. Integration of the Phase One system with NLR's Flight Management Computer (with limited capability) and installation of the second EFIS DU,
3. Integration with Navstar GPS,
4. Integration with MLS,
5. Integration with SSR Mode S,
6. Provision of 4-D navigation capability.

Each project phase will be concluded with a short flight test program to demonstrate proper operation of the system.

The schedule for the realization of these phases is shown in figure 7. The first phase, that is currently under development, will be ready in July 1987, while the second phase, providing a limited FMC capability and full EFIS, is scheduled to be realized by April 1988. The integration with GPS, MLS and SSR Mode S will occur as indicated in figure 7. However, this schedule may be changed if future applications of the ART system require so.

5. FIRST PROJECT PHASE

The first phase in the realization of the ART-project consists of:

- installation of one EFIS Display Unit in the right-hand side of the Metro instrument panel for display of primary flight information,
- installation of the EFIS DEU and required sensor systems in the cabin,
- generation of sensor signals in the desired ARINC 429 format for presentation on the PFD.

The second DU, for the display of navigation information will be installed in the second project phase when the FMC will be available.

A diagram of the systems used in phase one is shown in figure 8.

In this phase the ROLM 1664 computer is used for conversion of the analog signals from the different sensor systems to the ARINC 429 signal format required by the EFIS. Rather than having a number of ARINC 700 series systems connected to their dedicated ports on the DEU, the ROLM computer provides one serial ARINC 429 output containing information from all the systems that is distributed at the front end of the DEU Junction Box over the respective ARINC input busses. The data words in the ARINC output are identified by the proper unique ARINC label according to the ARINC 700 series specifications.

As a Digital Air Data Computer (DADC) will not be available before phase two of the project, the ROLM computer also contains software for the calculation of altitude, indicate' airspeed (IAS) and vertical speed based on measured static and dynamic pressures.

As shown in figure 8 the Honeywell IRS (ARINC 704) does not require signal format conversion and can therefore be directly connected to the EFIS for displaying attitude (roll, pitch) and heading information.

The Primary Flight Display format that is used in the realization of phase one is depicted in figure 9 and features the full basic "T": attitude director indicator (ADI) with flight director bars framed with speed data on the left, altitude and vertical speed on the right and horizontal situation indicator (HSI) below. ILS glide slope deviation indication is sandwiched between ADI and altitude tape, with radio altitude just below it.

Table 3 lists the parameters used to drive this display with their ARINC label and source.

Figure 10 and 11 show the cockpit panel lay-out before and after the installation of the single DU. As can be seen the DU installation was accommodated to a large extent by relocating the instruments; it was especially important to retain the HSI, mounted on this side of the panel, as this instrument is used for track selection in this phase of the ART-project.

After the realization of this first phase, NLR is under contract to perform research with this system in the areas of EFIS display formats and MLS interception procedures. For this last application the system will initially be expanded with an accurate position determination system, simulating MLS until MLS air and ground equipment becomes available.

6 SECOND PROJECT PHASE

The hardware used in phase two is based on the phase one hardware with the addition of the second EFIS DU, DADC and cockpit CDU. The cockpit panel lay-out is shown in figure 12. The software development effort for the FMC has been started already and initial software has been created for:
- flight plan generation,
- position determination, and
- calculation of track deviation errors (both lateral and vertical).

A computer model has been developed for an IBM AT personal computer with monochrome and color graphics screens for the generation of a navigation plan (Flight Plan). Because of the I/O facilities available it is intended to perform most flight planning activities on this computer, the end result being a floppy disk or tape that will be downloaded to the ROLM 1664. Figure 13 shows an example of the use of this system.

For the in-flight position determination a 12-state Kalman filter has been designed based on an Inertial Sensor System and a -generic- position fix system. Furthermore, software has been developed for the processing of flight plan information and for the computation of differences between the desired flight track and actual aircraft position.

7. CONCLUSIONS

The technical development of avionics systems takes place at a high rate. However, the advantages to be gained from these systems can only be realized if safe and proper procedures for the use of these systems in relation to Air Traffic Control can be established. One of the activities required to support the establishment of procedures is airborne research.

Based on its flight test and simulator experience NLR has started the development of such an airborne avionics research facility. In the framework of the Avionics Research Testbed (ART) project NLR's Metro research airplane is being equipped with a number of advanced avionics systems, including a programmable EFIS, programmable FMC, MLS, GPS and SSR-Mode S. The development of the first phase of the project, related to the installation of EFIS, is well under way and is scheduled to be finished by this summer, after which the system will be used in a number of projects. The total ART capability is scheduled to be available by 1989.

REFERENCES

1. Erkelens, L.J.J.; Geest, P.J. van der — Investigation on MLS approach path interception and transition techniques
Part II: Flight Simulator Investigation
NLR TR 85097 L, Amsterdam 1985.

2. Anon — ICAO Communications/Operations (COM/OPS) Divisional Meeting 1985
Montreal, 4-28 Sept. 1985.

3. Anon — International Standards, Recommended Practices and Procedures for Air Navigation Services.
Aeronautical Telecommunications.
Annex 10 to the Convention on International Civil Aviation. ICAO April 1985.

4. Anon — NATO Standardization Agreement Navstar Global Positioning System (GPS)
System Characteristics - Preliminary Draft
STANAG 4294, Draft Issue G, NATO unclassified.
6 Sept. 1985.

5. Anon — Secondary Surveillance Radar.
Mode S
ICAO Advisory Circular 174-AN/110. Montreal 1983.

TABLE 1
Comparison of MLS and ILS

OPERATING FREQUENCY	108 - 112 MHz (LOC) 328 - 335 MHz (GS)	5031 - 5090 MHz
NR CHANNELS	40	200
APPROACH PATH	ONE STRAIGHT IN	MULTIPLE, SELECTABLE SEGMENTED APPROACH CAPABILITY
ENVIRONMENTAL REQUIREMENTS	500 M FLAT AREA IN FRONT OF GS ANTENNA	NO SPECIAL REQUIREMENTS
MULTIPATH	PROBLEM	NO SIGNIFICANT PROBLEM
ANTENNA SIZE	LOC 15 -50 m WIDE GS 10 m HIGH	AZIM. 2.5 x 2.5 m ELEV. 3 m HIGH
DATALINK CAPABILITY	NO	YES

TABLE 2
GPS Accuracies

	PPS	SPS
Position	18 m (2 DRMS)	100 m (2 DRMS)
Altitude	28 m (95%)	157 m (95%)
Velocity	0.1-0.5 m/Sec	0.5 m/Sec
Time	180 nsec	385 nsec (95%)

TABLE 3
EFIS input parameters for ART Phase 1

nr.	parameter	ARINC source-label	a/c source
1	pitch	704-324	IRS
2	roll	704-325	IRS
3	track angle	704-313	IRS
4	magn. heading	704-320	IRS
5	baro corr. alt.	706-204	A/C
6	ind. airspeed	706-206	A/C
7	rate of climb	706-212	A/C
8	VOR bearing	711-222	A/C
9	VOR sel. radial	711-100	A/C
10	DME distance	709-202	A/C
11	ADF bearing	712-162	A/C
12	ILS loc. dev.	710-173	A/C
13	ILS GS dev.	710-174	A/C
14	radio altitude	707-164	A/C
15	Flight Director roll	7^1-140	A/C
16	Flight Director pitch	.01-141	A/C
17	barom. setting	706-234	ROLM
18	decis. height	701-170	ROLM
19	selec. heading	701-101	A/C
20	max. all. airsp.	706-207	ROLM
21	selec. airspeed	701-103	ROLM
22	selec. altitude	701-102	ROLM
23	sel. runw. head.	701-105	A/C
24	outer marker	701-222	A/C
25	middle marker	701-222	A/C

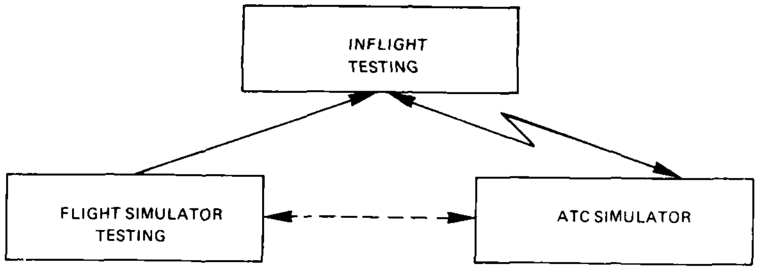
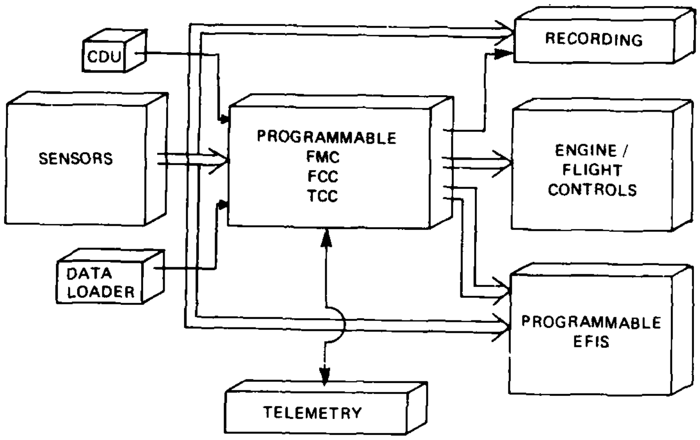


Fig. 1 Operational Avionics Research at NLR

Fig. 2 NLR Metro II Research Aircraft



FMC FLIGHT MANAGEMENT COMPUTER
FCC FLIGHT CONTROL COMPUTER
TCC THRUST CONTROL COMPUTER

EFIS ELECTRONIC FLIGHT INSTRUMENTS

Fig. 3 Art Concept

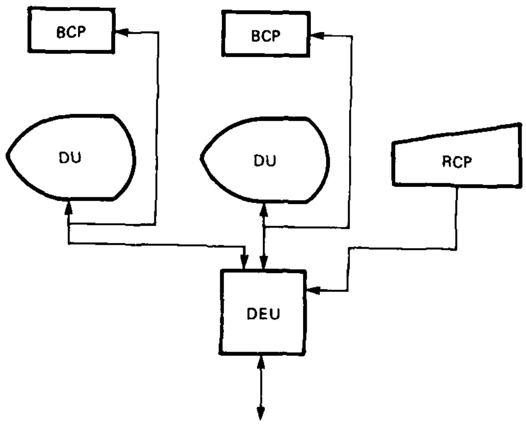


Fig. 4 EFIS Color Display System.

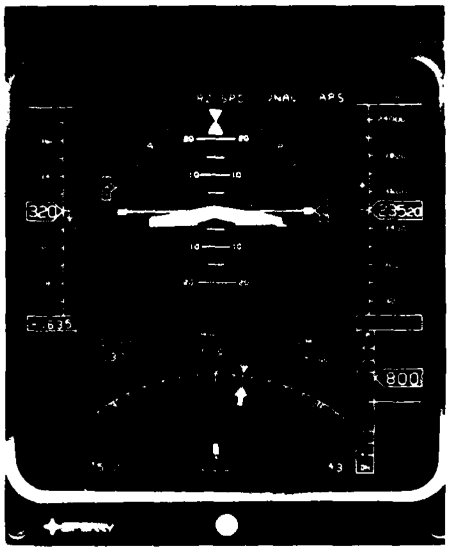


Fig. 5a: Primary Flight Display

Fig. 5b: Navigation Display

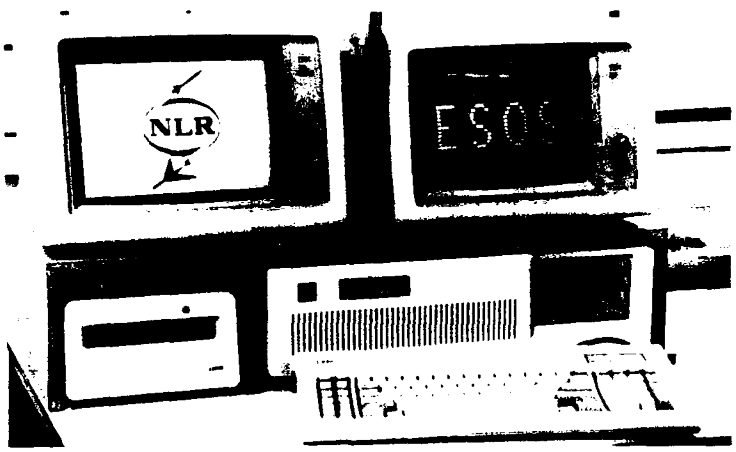


Fig. 6 Picture of EFIS Software Development Stat'on

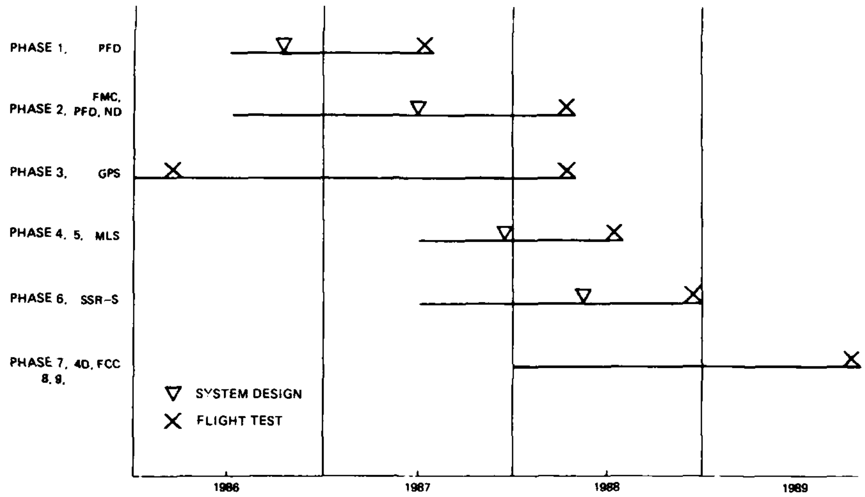


Fig. 7 Planning ART Project

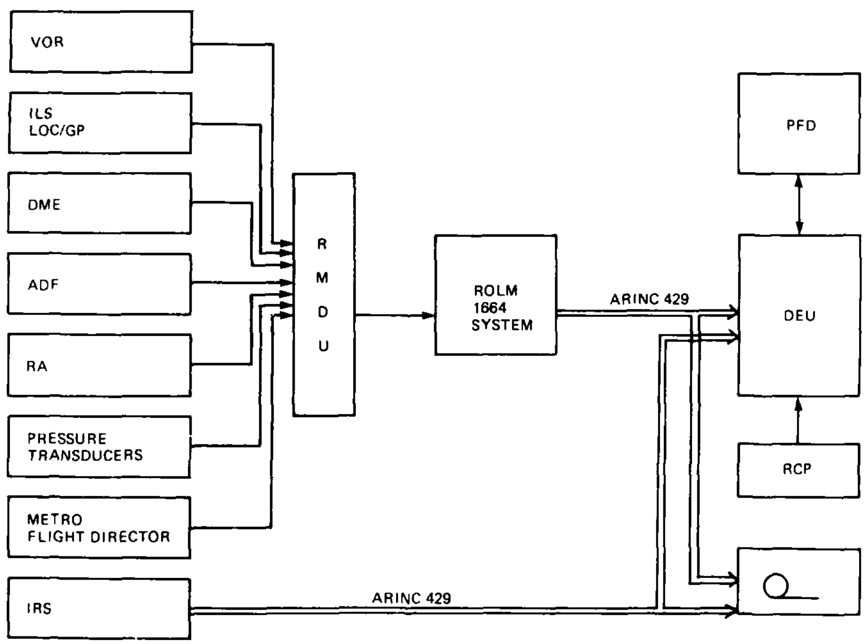


Fig. 8 Main Systems Art Phase 1

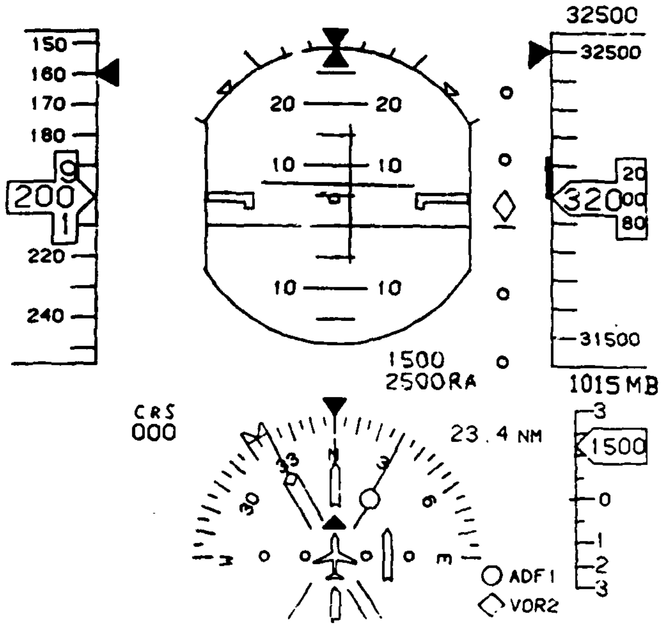


Fig. 9 Primary Flight Display Format used in ART Phase 1

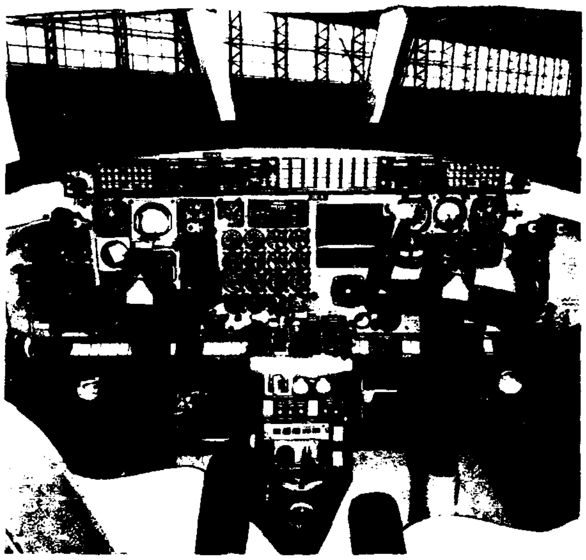


Fig. 10 present Metro Instrument Panel Layout

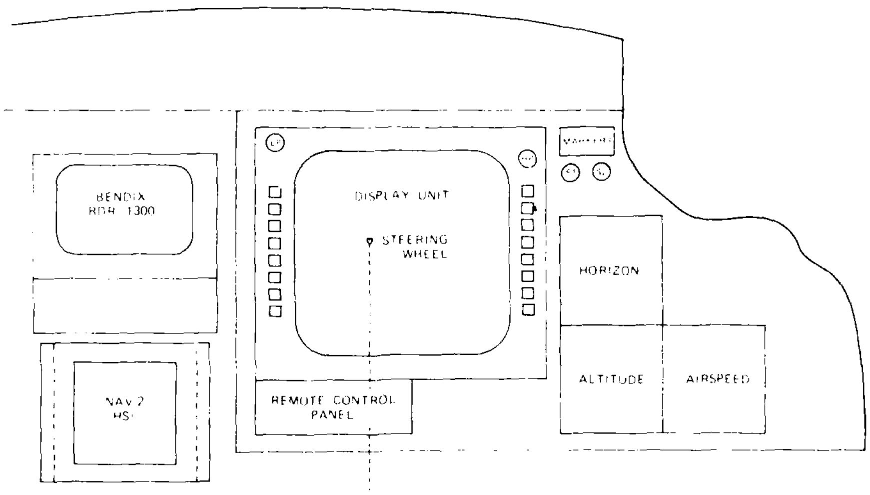


Fig. 11 Installation of single EFIS Display unit

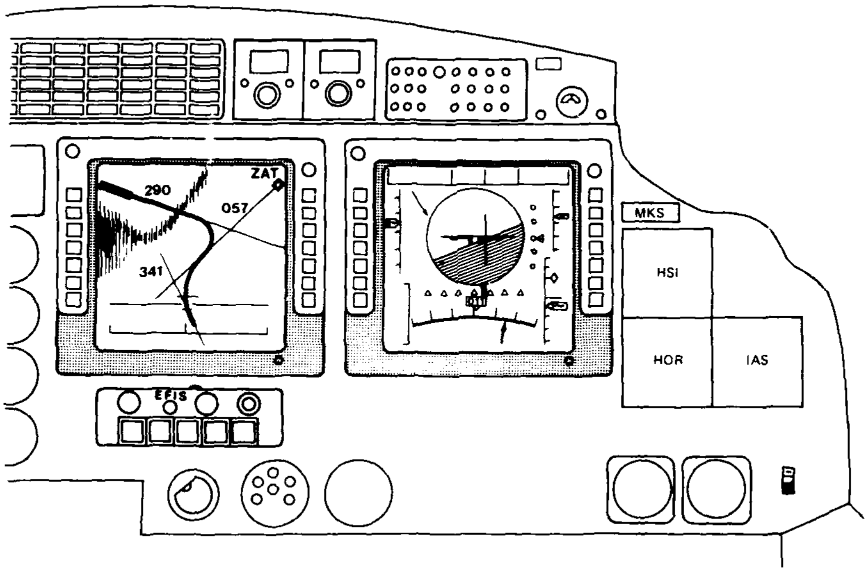


Fig. 12 Instrument panel Phase 2

Fig. 13 Example Of A Flight Plan

DISCUSSION

W.R.Fried, US

(1) Do you have any comments on the incompatibility between the altitude measured by GPS (above the geoid) versus the altitudes needed for air traffic control (i.e., barometric or pressure altitudes)?

(2) Could GPS time be used for the fourth dimension in air traffic control in the future?

Author's Reply

(1) GPS provides very accurate horizontal positions. Vertical accuracy, however, is not as good. Civilian users may obtain $\approx$ 150 m 1 σ in altitude. This does not provide much advantage over barometric altitude.

(2) It certainly can, but it may be an overkill in accuracy. Also, then everybody would have to use GPS to operate in the same time frame.

ATELIERS DE CONCEPTION DE SYSTEMES AVIONIQUES ET DE REALISATION DE LOGICIELS EMBARQUES

par

Monique Slissa
Aerospatiale Division Hélicoptères

et

Philippe Laroche-Levy
Avions Marcel Dassault
Quai Marcel Dassault
92214 St Cloud
France

La qualité globale d'un système avionique complexe dépend largement des choix de principe qui doivent être faits tôt dans le cycle de développement. Ceci justifie l'emploi systématique de moyens adaptés dès ce stade et tout au long du cycle de vie.

Comme support aux méthodologies de conception et de réalisation de systèmes qui sont présentées dans l'exposé cité en référence 1, il est donc nécessaire de mettre en oeuvre des moyens permettant d'élaborer et de maîtriser, tout au long du cycle de développement et de vie :

- La définiton d'un système

  C'est l'objet de l'ATELIER SYSTEME.

- La réalisation des logiciels embarqués

  C'est l'objet de l'ATELIER LOGICIEL.

Dans la suite du propos, les points suivants seront examinés :

- La démarche conduite par les avionneurs AEROSPATIALE et AVIONS MARCEL DASSAULT
- La notion d'atelier
- L'implantation industrielle des ateliers
- L'Atelier Système
- L'Atelier Logiciel
- Les communications entre ces ateliers

I - LA DEMARCHE CONDUITE PAR LES AVIONNEURS AEROSPATIALE ET AVIONS MARCEL DASSAULT

Pour le développement des systèmes avioniques de leurs précédents programmes aéronautiques, l'AEROSPATIALE et les AVIONS MARCEL DASSAULT ont utilisé des outillages informatiques, en particulier pour ce qui est de :

- la définition fonctionnelle des systèmes

- la spécification des traitements "temps réel"

- la gestion de configuration des systèmes et des logiciels embarqués

- la documentation associée aux étapes de conception et de réalisation.

Ces sociétés, sous l'égide du Ministère Français de la Défense, ont largement contribué à la définition de moyens performants couvrant le cycle de conception d'un système et de réalisation des logiciels embarqués qui y sont inclus.

Comme cela a été expliqué dans l'exposé cité en référence 1, ces travaux ont été conduits en collaboration avec la communauté avionique française, dans le cadre de l'étude ITI à laquelle prennent part, avec AEROSPATIALE et AMD-BA, les sociétés suivantes : CROUZET, Electronique Serge DASSAULT, SAGEM, SFENA, SFIM et THOMSON-CSF.

Ils ont abouti au développement de deux entités conformément à un processus de conception descendant :

- un ATELIER SYSTEME et

- un ATELIER LOGICIEL.

Ces ateliers doivent, en outre, répondre aux besoins de communication entre les activités qu'ils supportent respectivement. Ceci doit permettre de garantir la cohérence de cet ensemble qui constitue le Système de Développement d'Avionique ITI (SDA ITI).

Les développements du SDA ITI, actuellement en cours, ont pour objectifs la mise en oeuvre et l'utilisation de ces ateliers dans les prochains programmes aéronautiques qui impliquent des membres de la communauté industrielle de l'avionique française.

2 - LA NOTION D'ATELIER

Le terme d'"ATELIER", choisi pour désigner les deux grands ensembles composant les moyens d'étude et de développement de systèmes qui sont l'objet du présent propos, met l'accent sur des notions dont il faut souligner l'importance. Elles sont plus particulièrement relatives à la structure informatique de l'atelier.

- Chaque ensemble qualifié d'"ATELIER", comprend un certain nombre d'outils logiciels. *Ceux-ci peuvent être utilisés indépendamment les uns des autres et même en ignorant la structure de* l'atelier. En effet, ils sont conçus pour supporter les tâches relatives à une phase spécifique du développement d'un système. Il ne s'agit donc pas d'un ensemble informatique monolithique.

- Par ailleurs, chaque atelier n'est pas seulement constitué par la juxtaposition d'outils qui s'ignorent. Au contraire, chaque outil, même s'il peut fonctionner sans utiliser tous les autres services de l'atelier, est adapté :

  - Pour pouvoir recevoir des informations en provenance d'autres outils de l'atelier ainsi que pour en fournir lui-même, conformément à sa fonctionnalité et selon un processus d'enrichissement.

  - Pour permettre la préparation, le suivi et la gestion des évolutions des "produits" qui sont élaborés avec son aide dans le cadre d'un programme avionique.

Ainsi, un "ATELIER" est bien un ensemble d'outils logiciels dont l'un couvre la fonction particulière d'assurer l'intégration entre tous les autres. Cet outil appelé "STRUCTURE D'ACCUEIL", constitue ainsi la structure de base de l'atelier, à travers laquelle les utilisateurs accèdent aux outils spécialisés en gérant les activités liées aux autres outils et assure une gestion cohérente au sein de l'atelier, en particulier au sens de :

- la circulation des données entre outils

- *du suivi des évolutions des systèmes développés.*

A ce point, l'on peut ajouter deux remarques, qui sont conséquences des définitions données ci-dessus et concernent la réalisation informatique des ateliers :

- Les outils logiciels d'un atelier, attachés aux travaux spécifiques d'une phase d'étude ou de développement gèrent chacun les données qui sont propres à l'activité qu'ils sous-tendent. La structure d'accueil, de son côté, gère les informations liées à sa fonction, sans redondance avec les outils, mais en liaison avec eux.

- Comme cela a été dit, les outils logiciels doivent être adaptés au mode de fonctionnement intégré de l'atelier. Cela signifie que des outils existants peuvent subir l'adaptation nécessaire sans qu'il faille recréer de toutes pièces un outillage informatique offrant des fonctionnalités équivalentes. Certains outils du Système de Développement d'Avionique qui seront mentionnés plus loin, sont dans ce cas.

Ces deux points mettent en relief l'ouverture informatique au sens d'une compatibilité ascendante qui est le fait des ateliers qui vont être décrits.

3 - L'IMPLANTATION INDUSTRIELLE DES ATELIERS

Le découpage des moyens de développement de systèmes en "ATELIER SYSTEME" et "ATELIER LOGICIEL", munis *des capabilités de communication nécessaires dont la nature sera explicitée plus loin, correspond à celui des* tâches à faire par les industriels impliqués dans un programme d'avionique.

En effet, . Atelier Système supporte les travaux de conception et de spécification du systémier. L'Atelier Logiciel, quant à lui, supporte les travaux de réalisation des logiciels embarqués, principalement effectués par les industriels équipementiers, mais aussi dans certains cas, un avionneur pour les logiciels assurant la gestion du système, par exemple.

Sur ce point, l'on peut faire les remarques suivantes :

- L'implantation industrielle de ces moyens peut dépendre du type de coopération imposée, dans le cadre d'un programme d'avionique, à tous les partenaires (clients, avionneurs, équipementiers).

- Quelles que soient les relations établies entre les partenaires d'un programme et aussi chez le même industriel (si tel est le cas), les équipes de définition des systèmes d'une part et celles de réalisation des logiciels d'autre part sont par la nature même de leurs travaux, différentes et nettement séparées.

  Ceci justifie, en tout cas, la séparation des moyens en deux ensembles pouvant s'échanger les informations nécessaires à la conduite des travaux.

- Il apparaît évident, dans le cadre d'un programme d'avionique donné, que l'Atelier Système, permettant au *systémier de maîtriser la définition de l'ensemble du système, vis à vis de son client, doit être unique* ; ceci est également vrai même si, dans les faits, le "systémier" est composé de plusieurs industriels coopérants.

  Par contre, il est tout aussi clair, qu'au niveau de la réalisation des logiciels, la composante "Atelier Logiciel" est susceptible d'avoir plusieurs variantes, liées au contexte industriel et à la nature des logiciels à développer. Ce qu'il est important d'assurer, dans ce cas-là, afin de garantir la cohérence des développements dans le cadre du programme considéré, c'est

- Une méthodologie commune pour les divers Ateliers Logiciels,

- Des communications efficaces entre l'Atelier Système et chaque Atelier Logiciel.

4 - L'ATELIER SYSTEME

Il ne s'agit pas de reprendre ici une description détaillée de la méthodologie de conception et de définition d'un système avionique. Cela est exposé dans le propos cité en référence 1. Mais, celle-ci étant acquise, l'on va décrire précisément ci-dessous l'ensemble des moyens qui en constituent le support : L'ATELIER SYSTEME.

En premier lieu, seront énumérées les fonctionnalités attendues de l'Atelier Système ; ensuite les outils majeurs de l'Atelier Système ITI seront présentés ; enfin les caractéristiques les plus importantes de son implantation informatique seront indiquées.

4.1. Les fonctionnalités attendues de l'Atelier Système

Elles sont de plusieurs natures :

4.1.1. La mise en oeuvre de la méthodologie de développement d'un système avionique.

L'Atelier Système doit offrir des outils logiciels couvrant toutes les étapes du cycle de conception d'un système, chaque outillage informatique étant adapté aux travaux spécifiques d'une phase.

On peut rappeler ici, de manière macroscopique que les étapes principales sont les suivantes :

- élaboration des spécifications globales des fonctions opérationnelles

- détermination de l'architecture fonctionnelle

- superposition du découpage fonctionnel sur l'architecture physique

- spécifications détaillées des fonctions opérationnelles et des traitements effectués dans les équipements

4.1.2. L'obtention de spécifications de qualité

L'Atelier Système doit pouvoir être le garant d'un haut niveau de qualité pour les spécifications élaborées avec son support.

Ceci signifie, en particulier, que les outils logiciels doivent implémenter de manière rigoureuse les méthodes de travail préalablement choisies. Il faut, également, qu'ils permettent d'assurer un maximum de contrôles de cohérence à chaque niveau de spécification.

L'Atelier Système, surtout à travers les fonctions propres à la Structure d'Accueil, doit faciliter la prise en compte des spécificités liées à chaque programme d'avionique, comme la structure de la documentation, les circuits de décision et ceux de diffusion ..., tout en garantissant, après leur définition, leur strict respect pendant les travaux mettant en oeuvre l'atelier.

4.1.3. La maîtrise des évolutions de la définition d'un système avionique

La maîtrise des évolutions de la définition d'un système complexe implique en particulier la mise en oeuvre de moyens efficaces pour assurer correctement :

- La gestion des fiches d'évolution, élaborées au niveau global du système :
  - leur initialisation et la détermination de l'impact qu'elles ont au niveau des produits de l'atelier qui sont à la base des documents de spécification,
  - leur instruction à une profondeur suffisante pour permettre d'évaluer leur répercussion en coûts et délais,
  - leur prise en compte après une décision favorable.

- La gestion de configuration du système et la connaissance des applicabilités des fiches d'évolution et la définition des états, standards et versions du système.

4.1.4. La gestion de la diffusion des informations

L'Atelier Système doit prendre en compte les rapports entre les différents intervenants dans un programme d'avionique : les clients, les Services Officiels et Centres Etatiques concernés, le systémier, l'ensemble des coopérants ...).
A ce titre, il gère la diffusion entre les différents partenaires (documents de spécification, fiches d'évolution ...).

4.1.5. La mise à disposition d'un ensemble de moyens adaptés au niveau des utilisateurs concernés et au mode de travail requis par la méthodologie employée.

En effet, l'Atelier Système est destiné à être utilisé par des personnels de qualification élevée (chefs de projet, ingénieurs concepteurs, ...). Il doit donc présenter, dans ses différentes fonctions des interfaces "Homme-machine" adaptées. De plus, pour les phases de travail, menées au sein d'équipes de conception, les outils concernés doivent avoir des capacités "multiutilisateurs" puissantes afin d'assurer en permanence la cohérence de la définition.

4.2. Les outils majeurs de l'Atelier Système

Les outillages informatiques qui sont décrits ci-dessous font partie de l'Atelier Système inclus dans le Système de Développement d'Avionique ITI.

4.2.1. L'Outil de Conception Système OCS

a) Objet d'OCS

La spécification de l'architecture fonctionnelle d'un système est supportée par un outil d'analyse descendante, OCS.

L'objet principal de cet outil est de conduire à l'identification des pavés fonctionnels et au recensement des interfaces entre ceux-ci dans la démarche d'analyse fonctionnelle d'un système dont la complexité peut être variable. Cet outil supporte l'analyse de type IDEF0, mais a la souplesse nécessaire pour être utilisé en prenant en compte l'organisation du travail propre à chaque industriel.

b) Concepts manipulés par OCS

Les concepts pris en compte sont principalement les suivants :

- Les boîtes :

Elles constituent la représentation graphique d'un objet dont la sémantique correspond à la décomposition du système et qui représente une activité du système.

- Les flots :

Ils sont le regroupement sous un seul nom d'informations ayant une caractéristique commune. Leur représentation graphique est une flèche.

On rencontre donc les cas particuliers qui sont illustrés sur la fiche 1 :

. Un flot de commandes est un regroupement d'informations, contrôlant au sens d'IDEF0, l'activité représentée par la boîte à laquelle il est relatif.

. Un flot d'entrée est un regroupement d'informations utilisées en entrée de l'activité représentée par la boîte concernée.

. Un flot de sortie est un regroupement d'informations élaborées par l'activité de la fonction représentée par la boîte concernée.

. Un flot de mécanismes est un regroupement d'informations (ressources au sens d'IDEF0), permettant l'activité de la fonction représentée par la boîte à laquelle il est relatif.

- Les informations :

Ce sont les données transitant entre plusieurs éléments de la décomposition. Elles appartiennent toujours à un flot dont elles reprennent l'attribut (entrée, sortie, ...).

- Les dépendances fonctionnelles

Ce sont les dépendances entre les informations. Ainsi que cela est schématisé sur la figure 1, pour chaque information de sortie sont indiqués les liens avec les informations reçues permettant de l'élaborer.

c) Fonctionnalités d'OCS

La méthode d'analyse structurée hiérarchique descendante qui a été choisie (voir figure 2) permet :

. de commencer par la description la plus générale du système

. de déterminer ensuite les interfaces entre les différents éléments en raffinant la définition tout au long de la décomposition

. de terminer la décomposition en complétant l'architecture au fur et à mesure de l'avancement du travail par un processus itératif.

On procède donc à un découpage fonctionnel itératif, dans un contexte de multiutilisation approprié aux équipes de conception, qui assure en outre une propagation ascendante et descendante des informations.

Les principales fonctions de l'outil sont les suivantes :

. Des fonctions de création, modification et destruction de diagrammes et d'éléments de diagrammes permettant de saisir, de manière adaptée au mode de travail des utilisateurs, les données manipulées dans l'outil.

. Une fonction dite "configuration de l'outil" qui permet de déterminer pour chaque type d'usage, si cela est nécessaire, des paramètres d'utilisation, qui seront pris ensuite par défaut (comme les tailles maximales des noms de boites, de flots, ... ou les champs réservés aux attributs des informations ...)

. Des contrôles assurant la cohérence des travaux réalisés sous l'outil. Ils peuvent revêtir divers modes libre ou assisté, en temps réel ou différé.

. Des aides au travail en multiutilisation qui instaurent des dialogues intelligents entre les utilisateurs d'une même équipe de conception et permettent une utilisation simple des éléments de la décomposition spécifiés par d'autres.

Il s'agit en particulier de pouvoir mesurer en "temps réel" l'impact d'une modification et l'ampleur de sa propagation le long d'une chaîne fonctionnelle.

. La constitution et la manipulation d'arbres et de sous-arbres :

Dans un but de récupérabilité, des fonctions d'accrochage et de décrochage d'arbres et de sous-arbres sont mises en oeuvre avec les vérifications de cohérence et de complétude nécessaires pour en rendre l'usage valide.

. La gestion des évolutions dans laquelle le but poursuivi est d'utiliser l'intelligence de l'outil pour préparer les modifications avec la connaissance des fiches en étude et formuler les états modifiés définitivement après la décision d'application relative à une "version" du système.

. Le couplage à la Structure d'Accueil Système.

d) Produits réalisés à l'aide d'OCS

Lorsque les travaux relatifs à la spécification de l'architecture fonctionnelle, supportée par OCS, sont terminés, l'on a obtenu :

. La décomposition du système en plusieurs niveaux correspondant à différentes catégories de modules fonctionnels

. La détermination des flots de données transitant entre les différents éléments de la décomposition

. La spécification détaillée des interfaces

. La définition des chaînes fonctionnelles du système.

Les produits en sortie d'OCS ont à la fois des composantes graphiques et textuelles dont l'assemblage constitue les documents de spécification fonctionnelle d'un système.

4.2.2. Les outils de spécification détaillée

4.2.2.1. DLAO (Définition de Logiciel Assistée par Ordinateur)

a)Objet de DLAO

Il s'agit, avec cet outil, de faire la rédaction des spécifications détaillées des fonctions opérationnelles et de spécifier les traitements qui y interviennent et les test associés.
L'orientation de l'outil vers la prise en compte des activités à caractère temps réel le prédispose particulièrement à une utilisation dans la définition des systèmes avioniques.

b)Concepts manipulés par DLAO

Les données d'entrée de l'outil sont constituées par les interfaces et les chaînes fonctionnelles du système définies avec l'Outil de Conception Système, dans une étape précédente.

La spécification est rédigée à l'aide d'un langage formel manipulant les cinq types d'objets suivants :

. Les traitements

Ce sont les tâches accomplies par le pavé logiciel à spécifier. La description se fait en utilisant des mots-clés (faire, séquence, tant que, si, alors, ...). Elle comporte notamment :

- l'introduction des données d'entrée

- l'identification des données en sortie

- la liste des évènements agissant sur le traitement

- les contraintes de réalisation qui sont imposées.

La définition d'un traitement exprime son comportement dynamique qui se traduit par exemple par l'affectation d'informations, le déclenchement d'évènements, ...

. Les informations

Ce sont les données opérationnelles manipulées par les traitements. Elles peuvent être élémentaires ou composées d'autres informations.

- Les évènements

Ce sont les faits dont l'arrivée aléatoire ou cyclique activent ou conditionnent un traitement.

- Les états

Ce sont les éléments statiques intervenant dans un traitement ou qui s'en déduisent.

- Les interfaces

Ce sont les représentations physiques des informations dont la description précise est indispensable pour les phases suivantes du développement.

c) Fonctionnalités de DLAO

Les principales fonctions de l'outil sont les suivantes :
- La saisie interactive et l'analyse des objets prenant part à une spécification, basée sur l'utilisation d'un langage formel

- L'analyse et l'exécution de requêtes relatives aux objets déjà mémorisés pour en obtenir une représentation graphique ou textuelle

- Les contrôles de cohérence appliqués aux spécifications

- Une structure documentaire permettant :
  - La définition de plans-types de spécification adaptés aux besoins des utilisateurs
  - L'élaboration des documents de spécification, respectant les plans - types définis préalablement

- La gestion des évolutions présentant des mécanismes de même nature que ceux pris en compte par OCS

- Le couplage à la Structure d'Accueil Système.

d) Produits réalisés à l'aide de DLAO

Chaque spécification est une suite de définitions d'objets et de leurs attributs avec les relations entre les objets. Elle fait l'objet d'un document dont la nature a pu être définie par l'utilisateur.

4.2.2.2. SAO (Spécification Assistée par Ordinateur)

a) Objet de SAO

L'outil SAO aide à constituer la spécification qui prend en compte les exigences vis à vis du matériel et celles spécifiques au logiciel d'un équipement (algorithme d'une loi de pilotage ou de guidage, par exemple).

b) Concepts manipulés par SAO

La spécification est constituée par un ensemble de planches, répertoriées en livres et chapitres.

Les objets sont principalement saisis et manipulés à l'aide d'un langage graphique dont l'éditeur fait appel à des bibliothèques de symboles graphiques, adaptées aux types d'application.

c) Fonctionnalités de SAO

Comme le montre la figure 3, les principales fonctionnalités de SAO sont les suivantes :

- La saisie interactive des spécifications à l'aide d'un éditeur graphique et de bibliothèques de symboles,

- La gestion d'espaces de travail pour les concepteurs, indépendants de l'espace officiel contenant les états consultables des spécifications,

- Les possibilités de modification, de suppression et d'adjonction de planches dans une spécification avec une gestion associée des conflits

- Le contrôle de cohérence au niveau d'une planche de spécification

- Le contrôle de cohérence au niveau de l'ensemble de la spécification

- La gestion des évolutions, devant mettre en oeuvre des mécanismes analogues à ceux des autres outils de l'Atelier Système

- La capabilité d'être interfacé avec la Structure d'Accueil Système.

d) Produits réalisés à l'aide de SAO

Une édition de l'ensemble de planches cohérentes pour former une spécification constitue le document (graphique et textuel relatif à une version de la spécification concernée).

4.2.3. La Structure d'Accueil Système

a) Objet de la Structure d'Accueil Système

C'est le support informatique qui permet au systémier de gérer :

- la logique de développement d'un système,
- les rapports entre tous les intervenants dans un programme d'avionique,
- les demandes d'évolution concernant un système.

Il s'agit en effet :

- de maintenir l'ensemble de la documentation de spécification à jour et de gérer sa diffusion,
- de pouvoir identifier à partir du niveau le plus global du système l'impact d'une demande d'évolution,
- de garantir en permanence la cohérence de la définition d'un système,
- d'offrir aux responsables de projet la possibilité de consulter les différents états de développement de l'ensemble des entités ayant part à la définition d'un système,
- de gérer les procédures d'acceptation relatives aux demandes d'évolution et les répercussions sur les produits constituant les spécifications.

b) Concepts manipulés par la Structure d'Accueil Système

La Structure d'Accueil Système s'appuie sur une base de données et rassemble des mécanismes qui ont pour but :

- d'assurer la cohérence des informations connues dans l'Atelier Système, dans le cadre d'un programme d'avionique,
- de gérer les accès et les activités liées aux outils de l'Atelier Système,
- d'assurer la bonne exécution des fonctions de communication entre outils de l'Atelier Système, vers l'Atelier Logiciel, vers l'extérieur du Système de Développement d'Avionique.

c) Fonctionnalités de la Structure d'Accueil Système

La figure 4 montre les grandes fonctions de la Structure d'Accueil Système et les liens qui existent entre elles.

- L'initialisation de projet :

  Elle permet l'installation d'un projet dans l'Atelier Système, quant aux prévisions relatives au plan de développement du projet et à la configuration nécessaire pour utiliser l'Atelier Système pour le projet.

- L'administrateur :

  Il a pour missions principales d'interpréter les commandes et les requêtes des utilisateurs et de gérer les accès.

- Le suivi des évolutions :

  Cette fonction concerne la gestion des fiches d'évolution et implémente les exigences qui ont été énoncées plus haut.

- La gestion de configuration :

  Cette fonction a pour rôle de gérer les états du développement d'un système et les transitions entre ces états.

- Les activités liées aux outils :

  L'ensemble des fonctionnalités couvertes ici concerne tous les aspects de liaisons entre la Structure d'Accueil Système et les outils de l'Atelier Système.

- La composition de documents :

  Elle permet de créer des documents complets à partir des "produits" élémentaires contenus dans les bases de données propres aux outils et d'adapter leur structure aux exigences de chaque programme.

- En corrolaire la gestion de la diffusion :

Elle s'occupe de la tenue à jour des listes de diffusion des documents de spécification et de la mémorisation des documents diffusés dans le cadre d'un programme et en particulier en ce qui concerne leur version et leurs destinataires.

d) Produits réalisés à l'aide de la Structure d'Accueil Système

A la demande des responsables de projet, des états :

- d'avancement du projet
- des spécifications
- de la diffusion des documents
- des fiches d'évolution prises en compte, en étude,

peuvent être générés.

4.3. L'implantation informatique de l'Atelier Système

Un souci très vif de portabilité a prévalu dans les choix informatiques de base faits pour la mise en oeuvre de l'Atelier Système.

Les outils logiciels qui ont été décrits sont affectés à des stations de travail, fonctionnant sous le système d'exploitation UNIX. Pour l'implémentation des outils, l'emploi du langage C est favorisé toutes les fois où cela est possible. Les premiers choix de stations de travail sont APOLLO et IBM 6150.

La Structure d'Accueil Système est implantée sur une machine hôte, connectée aux stations de travail. Les premiers choix sont VAX (VMS), connecté à APOLLO et IBM (VM/CMS) connecté à IBM 6150.

Comme cela a été expliqué plus haut, les outils, d'une part, la structure d'accueil d'autre part gèrent les données qui leur sont propres, dans des bases de données indépendantes et sans redondance. Dans ce cadre-là, et toujours dans un souci de portabilité, d'ouverture et d'homogénéïté, le système de gestion de bases de données relationnel ORACLE a été retenu pour tous les outils et pour la structure d'accueil et cela, bien sûr, pour les deux versions de matériel informatique.

5 - L'ATELIER LOGICIEL

Comme cela a été expliqué dans l'exposé déjà cité en référence, les industriels partenaires de l'étude ITI, qui par ailleurs possèdaient en propre des éléments d'Atelier Logiciel ont pris communément conscience que le développement du logiciel à vocation embarquée est une activité très liée à la capacité créative des personnes. Cette capacité, si elle n'est pas canalisée par des méthodes et des outils rigoureux, peut conduire à des résultats qui ne sont pas compatibles avec les objectifs d'une entreprise industrielle en matière de maîtrise des délais et de la qualité.

En ce qui concerne les travaux de conception, réalisation et tests des logiciels, des outils nombreux et performants sont déjà en exploitation au sein des différents industriels membres d'ITI.

Pour ces domaines, le principal souci des travaux est donc de proposer un ensemble cohérent, possèdant des liens étroits avec l'Atelier Système, et disponible pour l'ensemble de la communauté aéronautique, ce qui a conduit à la définition d'un Atelier Logiciel pour lequel sont présentés successivement ci-dessous :

- Les fonctionnalités attendues
- Les caractéristiques des outils au regard du cycle de vie logiciel
- La mise en oeuvre.

5.1. Les fonctionnalités attendues de l'Atelier Logiciel

Les principales fonctionnalités attendues d'un Atelier Logiciel sont les suivantes :

a) Il doit être le support efficace d'une méthodologie rigoureuse de développement de logiciels.

En effet, les caractéristiques des logiciels intervenant dans les systèmes avioniques sont très pointues. Elles nécessitent l'adoption et la mise en oeuvre de méthodes de définition, de conception, de réalisation et de tests stricts et fiables.

Les logiciels de ce type participent à la sécurité des sytèmes auxquels ils contribuent. Ils comportent en particulier la mise en oeuvre de traitements "temps réel".

b) Pour répondre aux besoins de divers programmes d'avionique, qui peuvent se dérouler en même temps, l'Atelier Logiciel doit être capable d'accepter des environnements de programmation relatifs à différents langages.

c) L'intégration des modules logiciels et les tests relatifs à leur validation doivent être facilités par des moyens de test, conçus en liaison directe avec les autres éléments de l'Atelier Logiciel qui sont utilisés pour générer les entrées de ces phases ultimes de la réalisation des logiciels.

d) L'Atelier Logiciel a également pour mission d'offrir des moyens permettant d'assurer la gestion de configuration des logiciels en développement et de préparer les livraisons de logiciels.
Il s'agit de gérer les états du développement à divers instants du cycle de vie du logiciel et les transitions entre ces états. Les fonctions à remplir sont donc principalement l'identification de la configuration des logiciels et le maintien de son intégrité par une gestion efficace de l'application des modifications logicielles initiées au niveau global du système et transmises depuis l'Atelier Système.

e) Une interface "Homme-machine" conviviale et adaptée aux utilisateurs de l'Atelier (concepteurs de logiciels, programmeurs ...) est requise pour rendre pleinement efficaces les services de l'Atelier Logiciel.

f) L'Atelier Logiciel doit pouvoir être interfacé à l'Atelier Système selon les critères qui seront explicités plus loin dans le paragraphe consacré aux communications entre Atelier Système et Atelier Logiciel.

5.2. Caractéristiques des outils au regard du cycle de vie logiciel

Trois étapes principales peuvent être distinguées dans le cycle de réalisation d'un logiciel, à partir de la fourniture des spécifications.

a) Les phases de définition et de conception

Les activités qui y sont associées sont principalement :

- L'identification des fonctions logicielles et de leurs interfaces
- L'élaboration de la structure "temps réel" et la définition des mécanismes associés
- La définition des tests de validation
- L'analyse informatique du logiciel à réaliser.

Des outils logiciels spécifiques supportent ces activités de manière séquentielle.

b) La phase de réalisation du logiciel

Les activités conduites pendant cette phase sont les suivantes :

- le codage
- la production d'exécutables
- les tests unitaires.

Les langages de programmation embarquables retenus sont dits de haut niveau.

c) Les phases d'intégration et de validation du logiciel

Elles mettent en oeuvre les types d'outils suivants :

- Editeurs de liens
- Générateurs d'application
- Outils de tests statique et dynamique.

5.3. La mise en oeuvre

L'orientation prise en matière d'Atelier Logiciel par les industriels membres du groupe ITI est de s'appuyer sur les réalisations de l'atelier ENTREPRISE elles-mêmes financées par le Ministère français de la Défense et de participer aux évolutions de cet atelier.

Les buts d'ENTREPRISE de seconde génération sont les suivants :

a) L'atelier doit fournir un cadre cohérent pour le développement d'applications logicielles en LTR3, en langage C ou en Pascal, en ADA, pouvant inclure des parties écrites en assembleur.

b) L'atelier doit pouvoir gérer la cohérence de produits tels que documents de définition ou de conception, programmes sources ou objets, documentation de réalisation, applications partielles ou complètes, bibliothèques, moniteurs, programmes ou fichiers de tests, ...

c) L'atelier doit être doté d'outils assurant une plus large couverture du cycle de vie logiciel que ceux de la première génération.

d) La gestion de configuration doit viser l'extension à de nouveaux objets gérés au sein de l'atelier permettant :

- la gestion de la documentation
- le contrôle et la gestion des modifications logicielles.

e) La gestion de projet, en rapport avec la base de données d'ENTREPRISE, sera réalisée par un outil permettant de planifier les projets et d'en assurer le suivi en coûts et délais.

La démarche du groupe ITI vise l'obtention d'un Atelier Logiciel adapté aux activités de ses membres, ayant une définition cohérente avec l'atelier ENTREPRISE de seconde génération et qui pourrait en constituer un surensemble entièrement compatible. Dans ce but, l'accent est mis particulièrement sur les points suivants :

- La définition des capabilités de communication avec l'Atelier Système
- L'outil de définition de logiciel qui doit pouvoir exploiter les spécifications de logiciels venant de l'Atelier Système
- L'outillage supportant les phases de conception globale et détaillée de logiciel qui doit supporter une méthodologie :
  - Privilégiant une approche descendante structurée
  - manipulant des concepts cohérents avec ceux utilisés pour la définition du logiciel
  - préparant de manière correcte l'utilisation de langages de programmation de haut niveau
  - formalisant des notions de programmation structurée.
- L'intégration dans l'Atelier Logiciel d'un outillage ayant pour but la mise au point et le test dynamique des logiciels.

  Les principales caractéristiques demandées à un tel outil sont les suivantes :
  - La formalisation et la standardisation des opérations de test
  - L'automatisation de ces opérations et l'exécution de tests de non-régression
  - Le test en temps réel et la non perturbation du comportement des programmes à tester
  - La description sous forme symbolique des contrôles
  - L'indépendance vis à vis des moyens de production
  - L'indépendance vis à vis des machines cibles
- La mise en oeuvre d'une gestion de configuration prenant en compte tous les constituants d'une application logicielle donnée et tous les produits qui leur sont associés, disposant des mécanismes convenables pour le suivi des évolutions logicielles.

L'orientation retenue au niveau des matériels informatiques dans un souci de portabilité est la mise à disposition de l'Atelier Logiciel sur des stations de travail travaillant sous le système d'exploitation UNIX, le langage de réalisation privilégié étant C.

6 - LES COMMUNICATIONS ENTRE CES ATELIERS

Les échanges nécessaires entre l'Atelier Système et les Ateliers Logiciels utilisés dans le cadre d'un même programme avionique sont identifiés en tenant compte du partage industriel des tâches.

De manière globale, on peut dire que

- de l'Atelier Système vers un Atelier Logiciel

  transitent les spécifications des logiciels à réaliser et les demandes de modifications à appliquer

- d'un Atelier Logiciel vers l'Atelier Système

  transite l'identification de la fourniture logicielle (configuration du logiciel livré, équipement concerné ...)

Pour réaliser des communications efficaces :

- Il faut qu'un certain nombre de concepts soit commun entre ceux manipulés par les outils de l'Atelier Système pour élaborer les spécifications des logiciels, et ceux qui sont dans les Ateliers Logiciels pour effectuer la définition proprement dite du logiciel et l'identification de sa configuration.
- Il est nécessaire que les actions d'importation et d'exportation relatives à un atelier se fassent au travers de sa structure d'accueil, dont alors, en particulier la fonction "gestion de la diffusion" doit être activée

- L'implémentation informatique des liaisons entre ateliers comporte trois aspects :

  a) Les procédures de communication

  Elles sont à respecter rigoureusement au cours des échanges afin de maintenir la cohérence et la sécurité des informations.

  b) La réalisation matérielle

  Elle dépend principalement des moyens utilisés par les partenaires du programme considéré et aussi de leur situation géographique relative.

  c) La confidentialité

  Son respect nécessite la mise en oeuvre des moyens et de procédures spécifiques.

7 - CONCLUSION

La première version du Système de Développement d'Avionique (Atelier Système et Atelier Logiciel) va être utilisée dans les prochains programmes aéronautiques auxquels prendront part les industriels de la communauté avionique française. Ces programmes en permettront ainsi la validation en grandeur réelle.

Compte tenu du point actuel d'avancement des travaux de mise en oeuvre, tout laisse à penser que les moyens résultant répondront aux objectifs initiaux.
Ils constitueront le support nécessaire pour formaliser les relations entre industriels coopérants dans un programme avionique et assurer par voie de conséquence une meilleure qualité à un coût moindre en mettant à profit les compétences et expériences acquises.

L'aboutissement des travaux engagés constituera un point de départ.
En plus des évolutions et compléments inhérents à tout produit logiciel, le souci d'évolutivité qui a guidé le choix des techniques de réalisation en faisant du Système de Développement d'Avionique un système ouvert, permettra :

- L'introduction d'autres outils couvrant des activités complémentaires au cours du cycle de développement, ou répondant à des contraintes spécifiques d'un utilisateur

- L'utilisation de techniques et de technologies informatiques nouvelles, telles la définition et la mise en oeuvre de systèmes experts

Référence 1 : "Système avionique. Méthode de développement et outils informatiques" (Philippe LAROCHE-LEVY - AMD-BA, FRANCE).

Fig :1

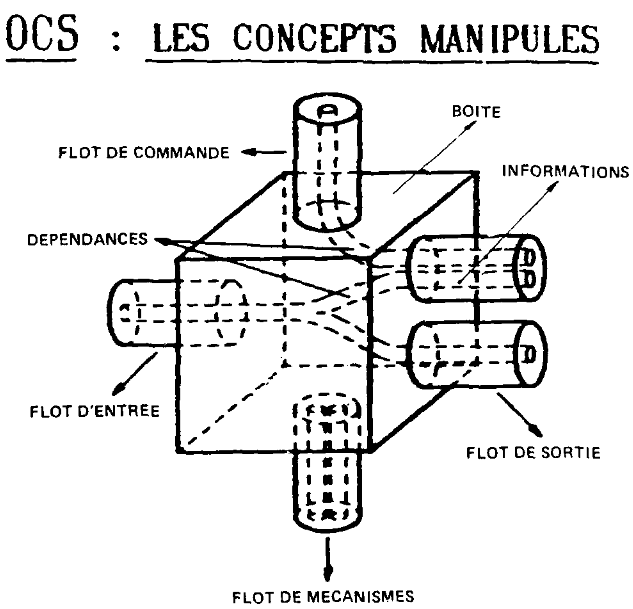


Fig :2 *OCS*

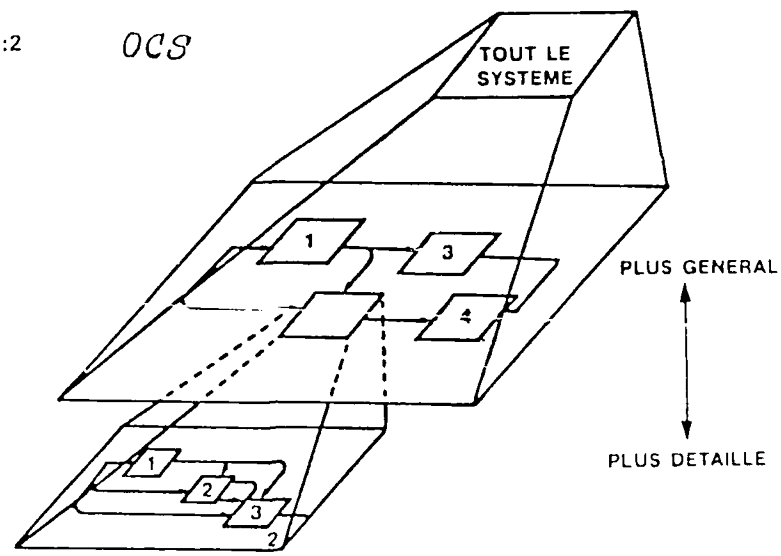
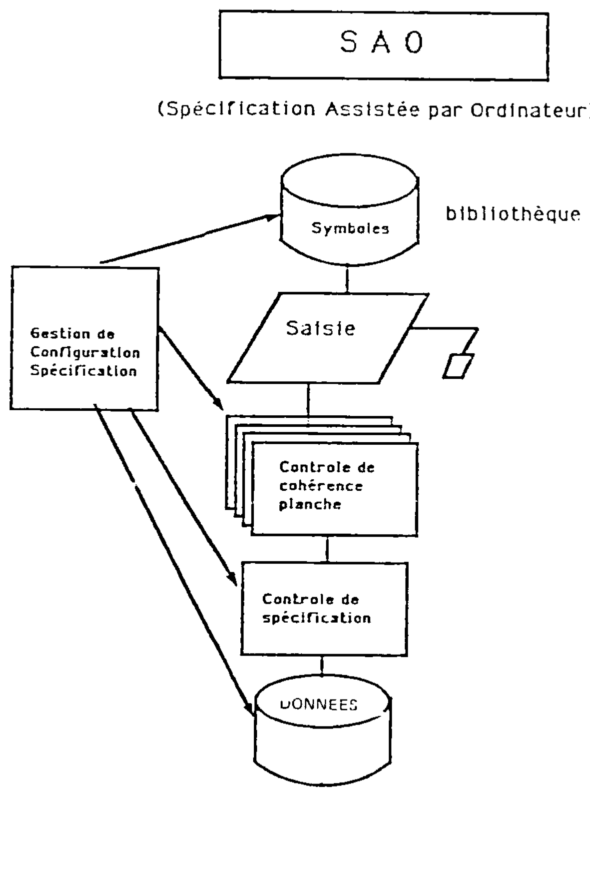
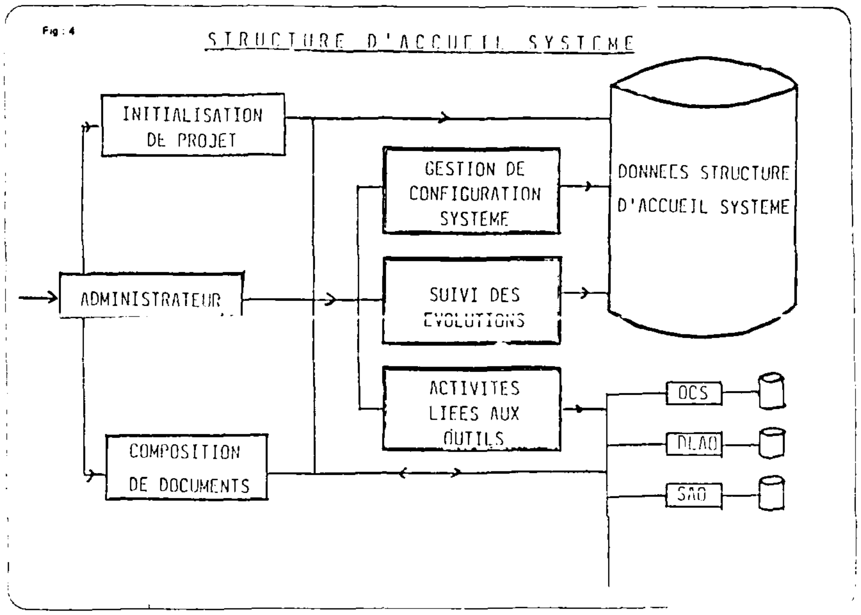


Fig : 3





R.Guiot, FR

Quel est l'accueil à ces outils par les utilisateurs?

Author's Reply

L'introduction progressive des outils auprès des utilisateurs est facilite par le fait que les outils supportent une méthode déjà appliquée dans les derniers programmes. Aussi, après une periode de formation, les utilisateurs decouvrent les gains attendus:

- Une meilleure communication entre les intervenants impliqués;
- Une aide à l'obtention de specifications de qualité, l'outil soulageant le specificateur de tâches repetitives de contrôle;
- Et principalement, une aide au maintien à jour.

COHERENT FUNCTIONAL DEVELOPMENT: KEY TO SUCCESSFUL FUTURE SYSTEM INTEGRATION

Bruce L. House
North American Aircraft Operations
Rockwell, International Corporation
2770 E. Carson St.
Lakewood, CA. 90712

ABSTRACT

An advanced, computer based engineering design methodology and tool set, which enables and enhances complex system, subsystem and component design, analysis, integration and verification/validation testing - in an operational context - is described here.

The burgeoning threat to the security of NATO alliance countries, dramatic escalation of costs and protracted schedules associated with weapons system acquisition, from concept development to first flight, dictates the introduction of drastic improvements in the way the technical community approaches the design and development process. Similar improvement must be made in techniques utilized by the procuring agencies. Improved methods for assessing the technical merit of proposed problem solutions as well as subsequent verification of compliance with procurement design specifications must also be established.

The timely evolution of enabling technology in areas of high speed digital computers, structured software, systems design methods, sophisticated simulation and graphics techniques have made these much needed improvements now possible.

Major failings of predecessor and even contemporary approaches to the application of computer-based simulation in the design process lie in: 1) relative inflexibility of the simulations, 2) persistent dedication to effectiveness analysis as an end-product rather than design optimization, 3) lack of deterministic assessment in favor of stochastic methods, 4) inadequate and/or esoteric means for data base access and manipulation, and 5) awkward, difficult to understand and use output data.

However, by far the most significant drawback of existing approaches to design, using computer based simulation tools, is that they fail to produce the coherent time-line state data essential to complete evaluation of the design objects (system/subsystem/component), in a dynamic, high fidelity and operational context.

When fully implemented, functional development by means of manipulation of coherently derived time-line state data, in a dynamic, high fidelity and operational context, will enable concurrent, fully integrated, rapid prototyping and subsequent detailed design synthesis of the avionics, vehicle, weapons and crew systems components of a total weapon system.

Having once put in place the described methodology and tool-set, opportunities for exploitation of the resulting coherent time-line state data are limitless, ranging from mission effectiveness analysis and detailed design, to test sets for logistic support and training devices, for both operations and maintenance.

INTRODUCTION

Central to our ability to bring the current protracted development cycle back into an affordable time span is the introduction of new and more powerful development tools and methods. Figure 1 illustrates our current posture of around twelve (12) years from CDI to IOC. This is nearly three (3) times that of only a decade or so ago.

The tremendous challenge precipitated by the "hypothetical" system requirements illustrated in Figure 1 creates a virtual log jam which may be portrayed by Figure 2. Elevated threat levels as well as rapidly evolving technology creates a much more difficult problem to solve and results in solutions featuring extremely complex design, development and test approaches. The end result is envariably protracted schedules, increased cost and uncertainty in performance achievability.

Contemporary weapons systems development approaches will not meet the challenge of the 1990's and needs help. A structured, formal and operationally relevant approach such as that alluded to in Figure 3 will be required. Traditional approaches fail in several respects. Firstly, each phase of the process generally employs stand-alone, unrelated and operationally non-relevant tool methods and procedures. Secondly, top level as well as flowed-down requirements are generally not derived in a dynamic operational context. This, of course, makes functional thread traceability virtually impossible. Figure 4 illustrates this traditional approach.

Advanced avionics systems are characterized by several important features which bear heavily upon the need for "requirements" to be developed and maintained (from mission level to the lowest level IPO and chip) in a dynamic operational context. First of all, advanced systems typically will contain $2 - 4 \times 10^6$ higher order language lines of code (Ada). This infers that many of the functional processes here-to-fore accomplished in hardware are now done in software. Secondly, processing speeds in the order of $3 - 4 \times 10^9$ complex operations per second (BOPS) for vector processing and $30 - 40 \times 10^6$ instructions per second (MIPS) will be required. Therefore, not only are most of the functional processes executed in a physically transparent and inaccessible media, but they are running at incredibly high speeds. Functional performance capability of a given system process, therefore, can only be assessed when operating in that high-speed, inter-active domain. Figure 5 itemizes some of the more important steps to be taken.

CONCEPT

A need is suggested by the foregoing for a methodology to "capture" the "design state" of each and every design variable involved in either the object of the design process or the "environment" to which the design object is subjected. Figure 6 depicts the process of coherently extracting the "state" data. The "state" data is referred to as "time-line" state data because it is developed and produced in response to some pre-determined time-ordered sequence of events.

To those who have labored rigorously over the past several years on advanced weapons system projects the need for "coherency" is self-evident. Quite simply stated, in an advanced weapons system (avionics suite specifically in this case) all components, i.e., sensors, busses, computers, etc., are utilized at all times when they can spatially, spectrally or temporally contribute to a functional process. Moreover, the components must continuously be assessed for availability as well. Data fusion (Radar/IR/CNI) is one example. Dynamic processor reconfiguration is another.

Figure 6 illustrates the occasion where "ownship" sensors are characterized and via digital simulation exposed to a threat environment under control of a simulation executive program. The response of "ownship" sensors in terms of their "design state" is collected and utilized for subsequent input into the core architecture computer, software and sensor design process. As shown, the product of the simulation design process is subsequently compared to actual hardware and software designs for requirements validity. Control of the design characteristics of "ownship" as well as cooperative Blue assets and Red threats is accomplished from the center block shown in Figure 6 as "Tool Management and Control". A non-real time, large scale version (80,000 lines of Fortran 77 code hosted on two micro-VAX) has been developed and is now in operational use.

Significant also is the fact that individual "functional threads" can be traced and controlled by this method as indicated by Figure 6.

Implied but not implemented in this version is the integration of both the Crew Station interfaces and Flight Controls interfaces (dashed lines at top and bottom of figure).

This early, non-real time coherent concept forms the basis for the major emphasis of this paper discussed in the following sections.

APPROACH

It is not possible to overstate the need for four (4) important factors to be embraced by any successful advanced development process:

1) Real-time operational context design and assessment.
2) Coherency between all design objects.
3) Careful sensitivity assessment between all design objects.
4) Computer based.

The object of this paper is a tool called "Coherent Design Evaluation Simulation (CODES)". Top-level features of the tool are illustrated by Figure 7.

Figure 8 illustrates the concept that a common environmental simulation which always accounts for the evaluation and response of all weapon system design objects can be implemented to achieve total weapons system functional and physical development (including real-time man-in-the-loop simulation and pilot vehicle interface/situation awareness simulations). Figure 8 is slightly misleading in that all of the models (avionics, vehicle, flight controls) are actually within "CODES" but are drawn as shown to illustrate their inter-actions.

When viewed in the context of the weapons system development process, as shown in Figure 9, the CODES tool can be used as the driver or data source for all development phases from mission effectiveness analysis through maintenance and training. All the necessary functional requirements are present at all levels, through all mission phases.

ARCHITECTURE

The software design of the CODES computer-based engineering methodology employs a high level of modularity in a hierarchial format for efficient upgrade, modification and maintenance. This architecture provides for efficient incorporation of modules representing various objects and sub-objects tc varying degrees of detail. Typical of the objects in CODES are airframes, radars, IR sensors, EW, missiles, guns, pilot, etc. The object

oriented approach allows for accommodating variable levels of detail within the methodology consistent with the objectives of the engineering design analysis. For example, radar system design analysis would use high levels of detail in those models. While other factors such as IRSTS, might be handled by first order models. Simulation fidelity versus run-time is an important consideration in virtually all applications of these types of methodologies. While the technical purist desires models, algorithm, and simulations to the Nth degree in detail, the pragmatist understands too well the impact upon computer run-time resource requirements, and ultimately the utility of the methodology. The methodology must be efficient to be useful and run-time is important. Obviously this factor becomes even more important when coupled to crew design synthesis tools and simulators in particular. The impact of inefficiency can be very expensive through the need for extensive computational facilities to support required simulator performance. Hence a balanced level-of-detail/fidelity versus resources trade-off is required if costs are going to be maintained within a manageable level. To partially achieve this, the methodology must be immersed within a "low-fidelity" context, to allow efficient utilization of "high fidelity" modules applied to the specific objective of study. This means putting the fidelity where it does the most good while using lower fidelity in areas which exhibit only second order effects.

Factors of a user friendly nature play a key role in the versatility of the CODES methodology. This aspect is of singular importance as it provides efficient operation, training, and simulation/methodology management with minimal operator expertise and effort. This aspect has been particularly important in the past as the operation and maintenance of a large number of data items in itself consumes extensive manpower resources. This consideration requires the application of modern software tools such as relational data bases, interactive menus, configuration agents, and a host of output display graphics which provide for flexible presentation of coherent time-line data. These latter elements, coupled with hard copy/presentation formats of data and graphics provide for maximum communication and transmittal of ideas, concepts, data, and trends, with minimal effort free effort.

The overall hierarchy of CODES is shown in Figure 10. It consists of an executive for interfacing the engineering software with the applications tools, i.e., crew and avionics design synthesis tools. The executive controls the data base management systems (DBMS), simulation models, and the post processors to provide formatted outputs to the applications.

Simulation Executive

The Simulation Executive provides the overall control for organizing, synthesizing, operating, and interfacing the computer methodology with various applications. It provides the mechanism for setting up the problem on an interactive basis with a minimal amount of manual overhead. It also assures that relevant modules communicate properly, synchronously developing the outputs to assure coherent time-line data. The executive routes these data for processing, graphic output and/or simulator interface.

Data Base Management System (DBMS)

The Data Base Management System (DBMS) provides the where-with-all to maintain and manipulate large amounts of weapon system design parameters. The CODES methodology is predicated upon evaluation of various system design options in an operational context. As such, not only are the engineering design parameters of interest stored in the DBMS, but so are parameters of all ancillary operational factors interacting with the methodology. As threat perception plays a key role, the organic variability and changing assessment of their design and performance factors must be easily updated in either manual or automatic (via data tape) fashion. Periodic update of threat

parameters occurs approximately every six months. Nominally 10,000 data items are required to define th performance parameters of the threat air defense system excluding geographic coordinates. The maintenance of these data is a critical task if relevant timely results are to be obtained.

In CODES, threat and engineering design parameters are maintained in a "bonded" data base with multiple password access. For a specific analysis/simulator run, these data are accessed and transferred to a run file where they may be used as is, or varied parametrically for purposes of engineering performance and sensitivity analysis. The DBMS configuration agent archives each runs data results so that future reconstruction of the analysis is possible. The flexibility of interactively modifying the run-file allows analytical exploration of various design alternatives, operations and tactics as well as long term assessment of threat growth implications.

Simulation Models

The simulation models in CODES develop the operation and performance of all the major elements of the weapon system and interacting environment for the conditions specified. These object-oriented models are of sufficient detail to permit sub-system design trades to optimize weapon system performance. These models represent all major avionic systems on the aircraft including radar, IRSTS, EW, IGNIA, etc. as well as the environmental factors that interact with weapon system operations, i.e., threat radars, missiles, C^3, GCI, clutter, visibility, etc. The executive links together these elements to construct somewhat generically the scenarios to be analyzed. For example, interactive menus provide prompts for selecting the number and type of red aircraft, blue aircraft, engagement geometry, scenario, key sub-systems of interest, tactics, operations, etc. The executive retrieves the relevant design parameters from the DBMS to convert the linked generic models to represent a specific engagement scenario. Fully specified, the simulation can be run over a broad set of engagement parameters to determine outcome. Obviously, the parameters can be exercised over excursions to identify sensitivity of system design to performance.

Post-Processor

The Post Processor performs data bookkeeping and develops graphic outputs for each engagement. Here the coherent time-line data generated in the system models and engagements is processed to develop measures of effectiveness. This output may vary from time-lines of design parameters, variables, and key events, i.e., target detection, missile launch, aircraft state, inventory draw-down; to generation of three-dimensional situational and synthetic cockpit displays (HUD'S C-SCOPE, etc.). This output is further processed and formatted to interface with the crew station synthesis tool. In this application, it provides the basic input for driving the various elements of the simulator.

MAN MACHINE/DESIGN TOOL INTERFACE

The Man Machine/Design Tool provides the interactive computer aided engineering design work station for purposes of performing system design trades in an operational context. With this methodology, the system engineer can synthesize an aircraft configuration including observables, offensive/defensive avionics, weapons tactics, performance, etc., and fly it through a hostile air defense environment. Furthermore, menu prompts provide the design engineer a training mechanism as well as continuous "help" functions to assist in data set-up, manipulation and engineering analysis. This methodology provides the system engineer the first evaluations as to the performance and shortcoming of his aircraft design from an avionics point of view. He can

observe via situation and synthetic cockpit displays how the engagement is progressing and what are the key performance drivers for maximizing system design. By varying design parameters, improvements in aircraft performance can be easily observed.

Approaches to aircraft system design have been exercised in a fragmented manner in the past. Each major sub-system designer has developed their requirements somewhat independently and in a piecemeal fashion. For example, the radar engineer would address the operational scenario differently than the weapons designer. This was often a laborious and time consuming process with full integration not occurring until the equipment was installed. Furthermore, the required coherent time-line data has been buried in a vast amount of analysis and test data with different assumptions and baselines used by the sub-systems developer. This made interpretation of results in an integrated fashion difficult if not impossible.

CODES utilizes the efficiencies of computer aided engineering methodology by providing an automated means of setting up and solving problems in a truly integrated fashion. The total aircraft configuration is synthesized in the computer, exercised in an operational context, optimized as an integrated design, and evaluated over a broad set of operational conditions. These evaluations can be conducted for both near- and far-term requirements to address an evolving advancing threat capability. The result is a set of coherent time-line data which provides state information on all major elements of the problem. This factor is key to developing optimized solutions for tomorrow's advanced aircraft system designs in a totally integrated manner.

REAL TIME MAN-IN-THE-LOOP INTEGRATION

CODES provides the core software for generating the reaction a hostile air defense system would undertake in response to an impending penetration. As such CODES Provides the methodology for driving a man-in-the-loop simulator, for simulating penetration of the hostile airspace.

Utilization of the core CODES software optimizes the utility of the tool and provides an efficient transition from engineering analysis to pilot-in-the-loop simulations. It is also important from the aspect of software maintenance and data base management. Further it not only allows efficient transition of variables from one weapon system design to another in the work station as well as the simulation, but easily accommodates changing threats and different operational scenarios. The work station and simulator can be easily "reprogrammed" by simply calling up the relevant menus and changing the appropriate engineering design parameters.

For example, suppose it is desirable to optimize the radar requirements versus an IR sensor. The engineering work station would be used to first perform system design trades between these sensors to identify performance overlaps and options. Then the sub-systems design modules for each would be exercised to verify the feasibility of designs to meet performance requirements. Once design baselines are established, engagements would be run inter-actively on the work station, and then the design data used to drive the simulator so that the vehicle can be evaluated with the pilot-in-the-loop.

HARDWARE/SOFTWARE

The utility of CODES can be extended even further for digital hardware-in-the-loop hot bench testing. An extension of CODES generates a bus of digital pulses as would be generated by various avionics sensors and processed by an ASA or PAVE PILLAR onboard the aircraft. As CODES generates coherent time-line data on all participants or the engagement, this data can be utilized to control "digital signal generators" and provide the proper signal bus format

to the digital processors. This feature allows the hardware to be tested over a broad range of test scenarios for validation and verification of processor performance.

Digital processors can also be tested by tying in with cockpit displays and indicators. Hence, the pilot-in-the-loop provides for effective evaluation of processor design through simulated flight runs in the simulator. With this configuration, data bus traffic, response format and timing can be tested and evaluated.

EFFECTIVENESS ANALYSIS

A fundamental role of CODES is mission effectiveness analysis. CODES provides the operations analysis the tool to efficiently evaluate vehicle mission effectiveness over a broad range of operational conditions and scenarios. The scope of CODES provides a full complement of Threats, C^3 environments, operations, tactics, etc. to allow synthesis of scenarios representing vast major operations. With CODES, the transition from work station, to simulator, to hardware-in-the-loop, to mission analysis represents a relatively straightforward process accomplished with menus and pointers through a totally interactive process. Hence the aircraft configuration can be optimized on the engineering work station, tested by a pilot-in-the-loop simulation, tested with digital processors-in-the-loop and ultimately exercised over a broad set of operational scenarios.

The utility of CODES in this respect is even broader as it provides a quick and efficient means of testing aircraft performance specifications against mission requirements. Further it allows integrated mission assessment from close air support to deep interdiction, identifying new or overlapping roles of advanced aircraft designs. Here the true value resides to the modern aircraft designer. For he must design his aircraft in a totally integrated manner to successfully compete and accomplish today's challenging requirements.

GENERAL CAPABILITIES

Figure 11 is typical of the CODES implementation process.

Proceeding from a set of requirements specified to the CODES tool via one of the many tools now under development, the design console shown is utilized to construct the mission scenarios and tactics statements appropriate to the missions as well as statements of the problem to be used as a basis for evaluating performance in terms of measures of effectiveness (MOEs).

The real-time mission scenarios are now executable, under simulation executive control, so that the design approach effectiveness can be observed and optimized. Figure 12 further illustrates important operational features of the CODES tool. Capabilities illustrated are real-time dynamic intervention capability of the operator, availability of time-line state data for on-line or off-line use by associated computer based analysis tools and on-line performance assessment of design performance. Some of the more salient features of CODES including why it is unique, are listed in Figures 13-(a) through (e).

Probably the most significant breakthrough presented by the CODES concept is the integration of the actual design process with real-time man-in-the-loop simulation. Through this integration process the designer can observe on-line, inter-actively and dynamically the effects of changes in the state of design variables. More importantly, the effects of simultaneously altering

the design of several design variables can be observed for impact upcn total weapon system performance.

The CODES tool will evolve through several levels ranging from all-digital simulation to incorporation of actual hardware and software. Figure 15 illustrates CODES Level 1 (all digital) with the heavy arrows indicating points of primary operator control. Figure 16 shows a heavy darkened line around the "Signal Processor Hardware" indicating incorporation of actual hardware or a hardware emulator. Figure 17 similarly illustrates incorporation of actual "Data Processor Hardware" (this infers, of course, the actual operational flight programs would be operating). Finally, Figure 18 illustrates the integration of CODES with a real-time, man-in-the-loop simulation. Important to note that the CODES "Level" is neither indicative of time phasing nor is it homogenous as sequenced. For example, real-time, man-in-the-loop simulation integration prior to completion of many other CODES design features is planned in the earliest phases.

One of the more important characteristics of the common concept, in addition to providing high-fidelity rapid prototyping capability at the systems, sub-systems or component level, is its software reconfigurability. As illustrated in Figure 19, for instance, the CODES tool is rapidly reconfigurable to address a wide variety of weapon system requirements through simple construction of data files, scenarios and unique algorithms.

Additional capabilities planned for the CODES tool include automatic software code generation and validation as well as AI algorithm development.

SUMMARY AND CONCLUSIONS

Protracted development schedules and escalating costs can be, at least in large measure, contained through utilization of advanced methods and tools. CODES represents but one offering of the type of tools necessary to do the job for reasons identified in Figure 20. The methodology must be pervasive throughout the many company disciplines (procurement, manufacturing, human resources, finance) for the development process to be truly speeded-up. Many companies are well on their way in most areas except the design areas addressed by this paper.

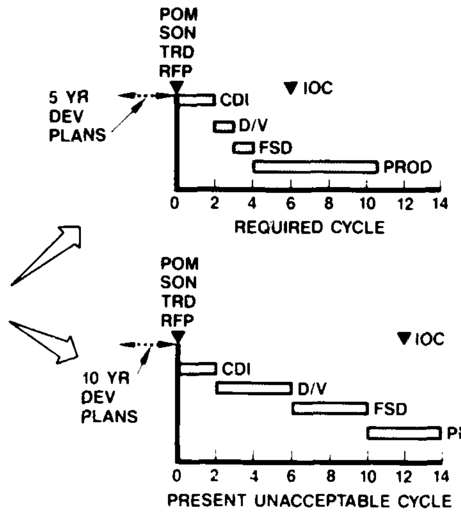
The impact of technological credibility upon social acceptance of defense projects becomes larger each year. For programs vital to our collective defense posture to be accepted requires high levels of technical credibility and fiscal responsibility. Only then will we again widen the narrowing gap between our alliance and the burgeoning threat.

(U)

Do We Have a Development Option?

1990
HYPOTHETICAL WEAPON SYSTEM
REQUIREMENTS

- Hi Mach sustained supersonic cruise
- Long combat radius
- Hi sustained G's
- Minimum TOGW
- Minimum crew
- Maximum payload
- BVR and CIC capable
- Extreme low observables — 4π
- Passive sensors across band — 4π
- Active sensors across band — 4π
- Day/night/in weather targeting
- Autonomous/co-operative capability
- Operable away from M.O.B.
- Highly reliable/maintainable and supportable
- Highly affordable
- Technologically transparent

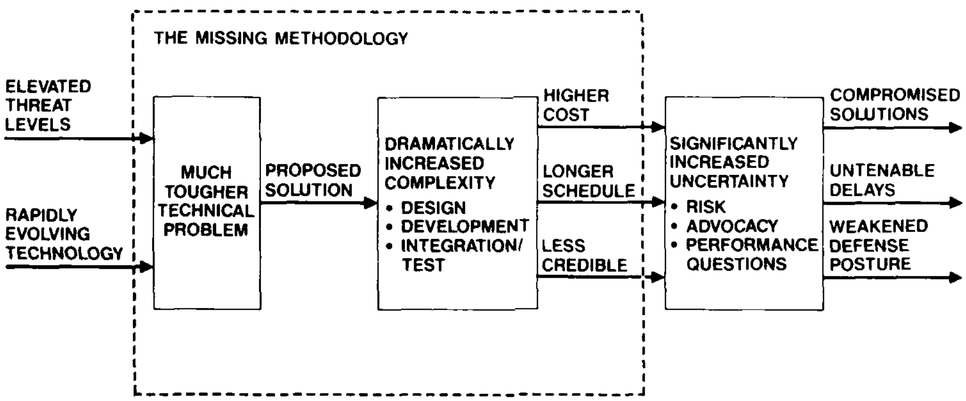


(U)

FIGURE 1

(U)

Breaking the Log Jam

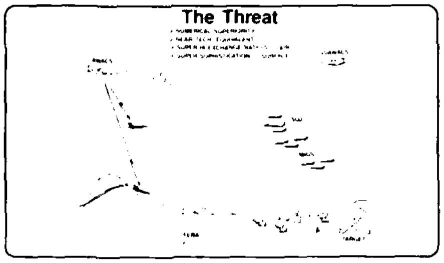


(U)

FIGURE 2

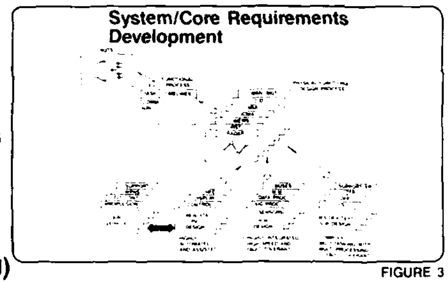
(U)

The Development Challenge



- Lethal
- Survivable
- Affordable
- Reliable
- Maintainable

- Coherent functional development
- Functional decomposition
- Close coupled interactive design development integration and test process
- Requirements traceability thru V&V and support
- Designed in technological transparency F31

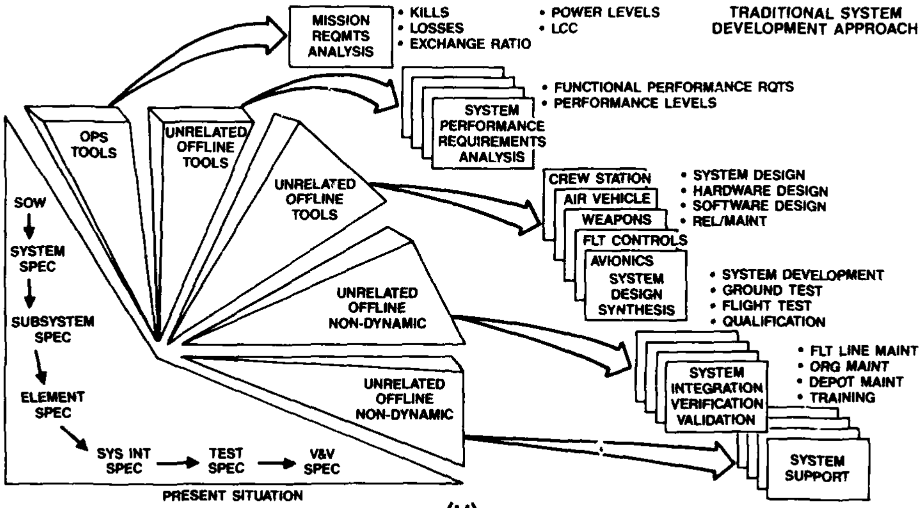


(U)

FIGURE 3

(U)

This Approach Needs Help



(U)

FIGURE 4

(U)

Help Must Come in the Form of ---

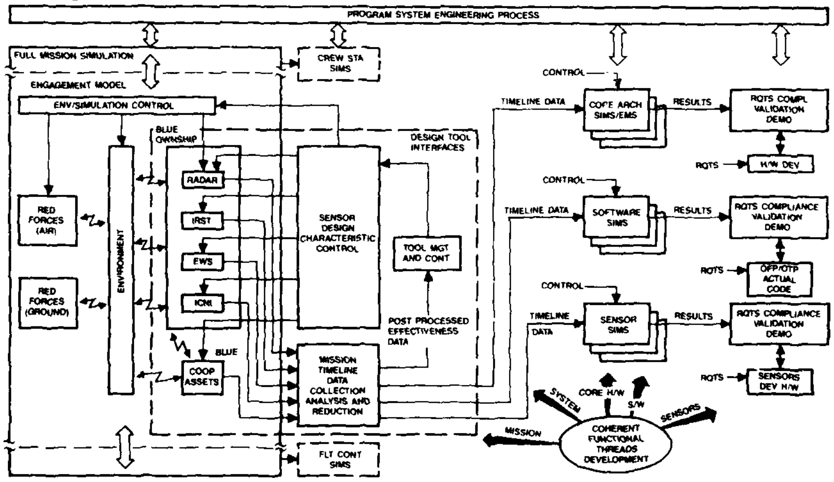
- New coherent methodologies for
  - Requirements analysis
  - Design synthesis
  - Design verification/validation
  - Integration and test
- Methodologies wrapped around computer based tools
- Automated, computer based project management and control
- Reduced levels of hi-tec staff
- Training programs to upgrade staff
- Organizational re-alignments
- Insight into requirements for long lead time capital investment to meet the challenge of the 1990's and beyond

(U)

FIGURE 5

(U)

Application of System Development Tools



(U)

FIGURE 6

(U)

Coherent Design Evaluation Simulation (CODES)

- CODES is an advanced computer based engineering tool to enable and enhance

  Design analysis and rapid prototyping

  Dynamic hi fidelity real time man-in-the-loop simulations

  Hardware-software system integration "in the dynamic environment"

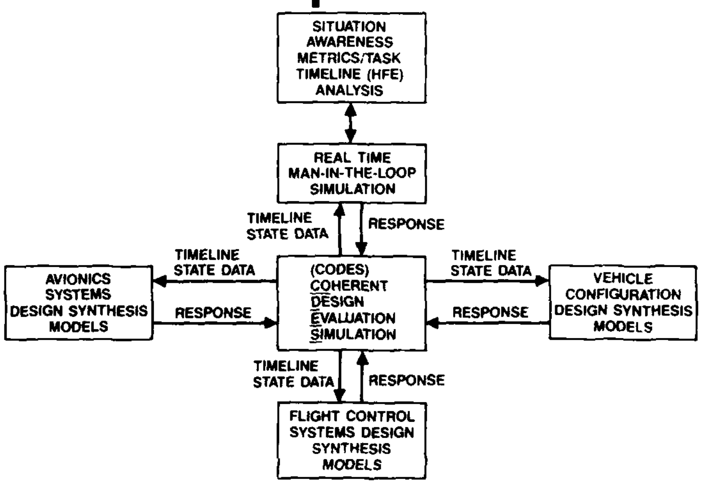
  Complete system design, development and integration capability

(U)

FIGURE 7

(U)

Coherent Design Evaluation Simulation Concept

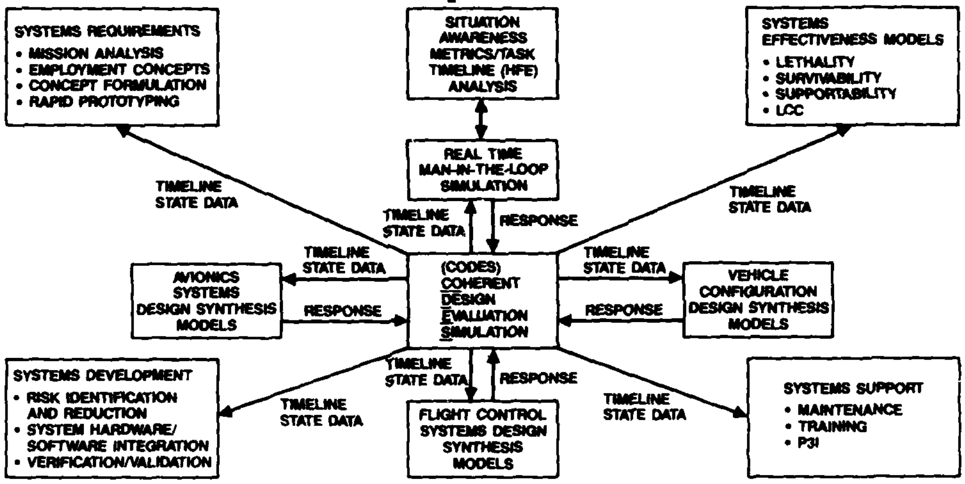


(U)

FIGURE 8

(U)

Coherent Design Evaluation Simulation Concept

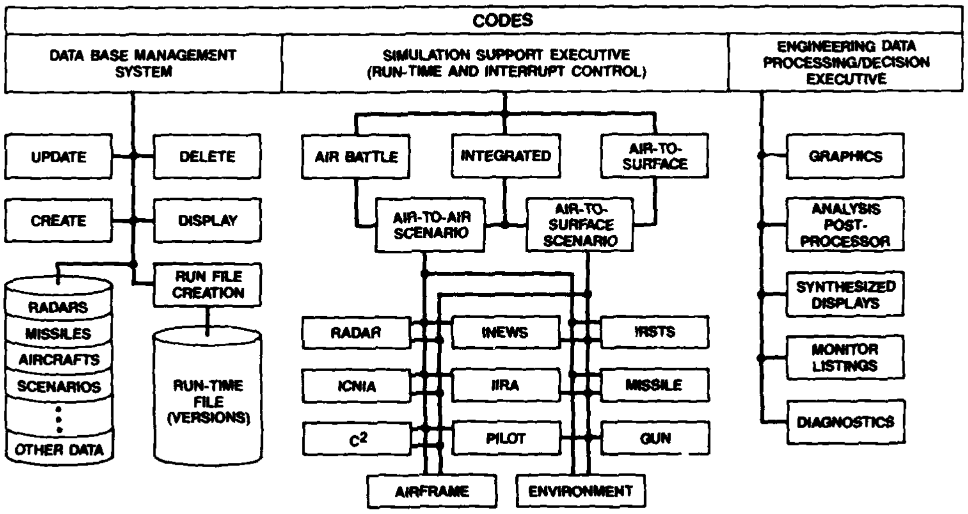


(U)

FIGURE 9

(U)

CODES Functional Hierarchy

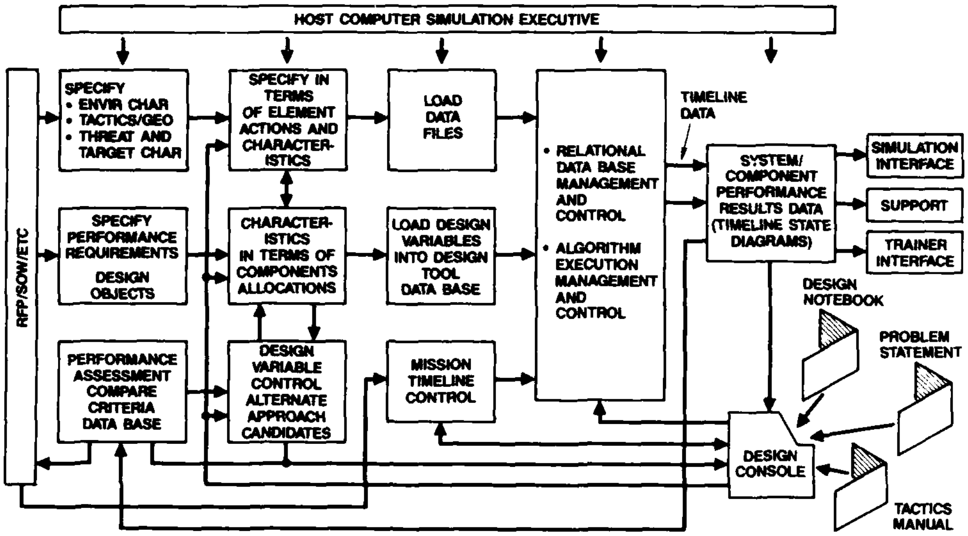


(U)

FIGURE 10

(U)

CODES Process

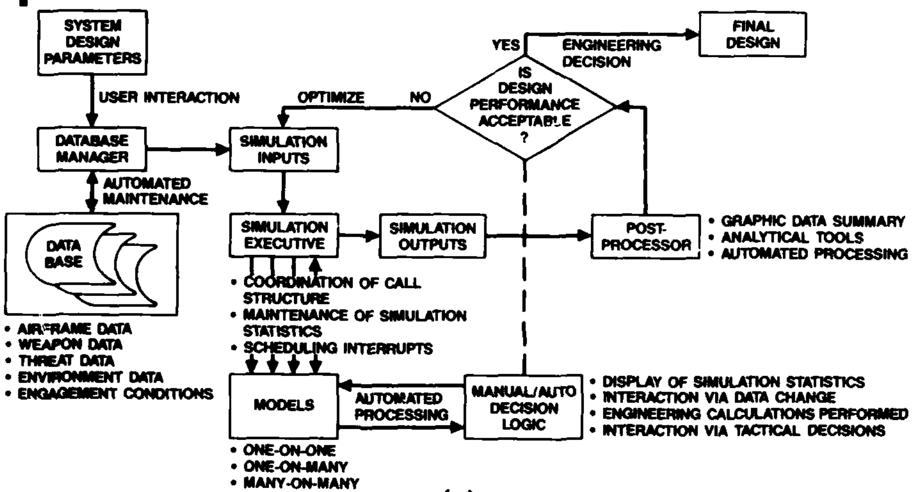


(U)

FIGURE 11

(U)

System Design Tool Operation Overview



(U)

FIGURE 12

(U)

What Makes Codes Unique?

- Integrated end-to-end penetrator/air defense engagement simulation
- Fully stand alone and menu driven
- Turn-key operation, user-friendly interface
- Automated data management of thousands of unique input parameters
- Modeling of all major avionics subsystems
- Automated model adjustment to selected scope of simulation
  - Air battle
  - Air-to-surface
  - Integrated
  - One-on-one
  - One-on-many
  - Many-on-many

(U)

FIGURE 13a

(U)

What Makes Codes Unique? (Continued)

- User interaction with simulation
  - Interrupt and display of simulation state
  - Parameter adjustment/optimization during simulation
  - Tactical decision making
- Custom graphic displays
  - Pilot displays
  - Situation displays
- Automated analysis tools
  - Data reduction
  - Design effectiveness measures
  - Graphical representations

(U)

FIGURE 13a (CONT'D)

(U)

Simulation Executive Module Scope

- Menu driven
- Variable I/O and data recording
- Interrupt scheduling
  - Time
  - Event
- Interrupt handling/manual decision logic
  - View pilot displays
  - Modify data
  - Perform engineering calculations
  - Plot scenario
  - Make tactical decisions

(U)

FIGURE 13b

(U)

Simulation Executive Module Scope (Continued)

- Coordination of players/scope of simulation
  - Air-battle
  - Air-to-surface
  - Integrated
  - One-on-one
  - One-on-many
  - Many-on-many
- Error checking and error handling
- Interrupt logging/data modification logging
- Event listings

(U)

FIGURE 13b (CONT'D)

(U)

System Models Scope

- Airframe
  - One or many
  - Blue and red
- Avionics
  - IRSTS
  - ICNIA
  - IIR
  - INEWS
  - AI radar
- Missile
  - One or many
  - Blue and red
  - IR, semiactive, command
  - Ground and air launched
- Gun
  - AAA
  - One or many
- Threat radar
  - One or many
  - Networked
  - Airborne and ground based
- C^3 network
  - Blue and red
- Environment
  - Terrain
  - Atmospherics
  - Visibility

(U)

FIGURE 13c

(U)

Data Base Manager Module Scope

- Menu driven
- Automated input and maintenance of data
- Data display options
- Two level data base
  - Data file data base — system data

$$\left.\begin{array}{l}\text{Airframe parameters}\\ \text{Missile parameters}\\ \text{Gun parameters}\\ \text{ADS site laydown}\\ C^3\text{ network}\\ \text{IRST parameters}\\ \text{ICNIA parameters}\\ \bullet\\ \bullet\\ \bullet\end{array}\right\}\text{Types of data files}$$

(U)

FIGURE 13d

(U)

Data Base Manager Module Scope (Continued)

- Two level data base (continued)
  - Input file data base — engagement specs
    Which aircraft
    Which weapons
    Which avionics
    Initial conditions
    Environment
    } Integrated data files
- Error checking and error handling
- File protection with passwords

(U)

FIGURE 13d (CONT'D)

(U)

Post-Processor/Analysis Module

- Menu driven
- Error checking and error handling
- Data displays
  - Threat laydown and flightpaths
  - Aircraft signatures
  - Site masking
- Simulation output displays
  - Detection contours (range vs angle)
    IR and radar burnthrough
    With and without jamming
  - Vulnerability contours (range vs angle)
    With and without maneuvers
    With and without jamming
    With and without expendables
    } Self-protection effectiveness

(U)

FIGURE 13e

(U)

Post-Processor/Analysis Module (Continued)

- Simulation output displays (continued)
  - P_K and shot opportunities
  - Weapon lethality
  - RWR signal history/sensitivity
  - Synthesized displays (snapshot)
- Situation plots
- Analytical tools (outputs)
  - Sensor data correlation
  - Signal/data processor loading histories
  - C^3 network loading histories
  - Exchange ratios
  - Weapon delivery accuracy
  - Target kill probability
  - Targets/penetrators missed

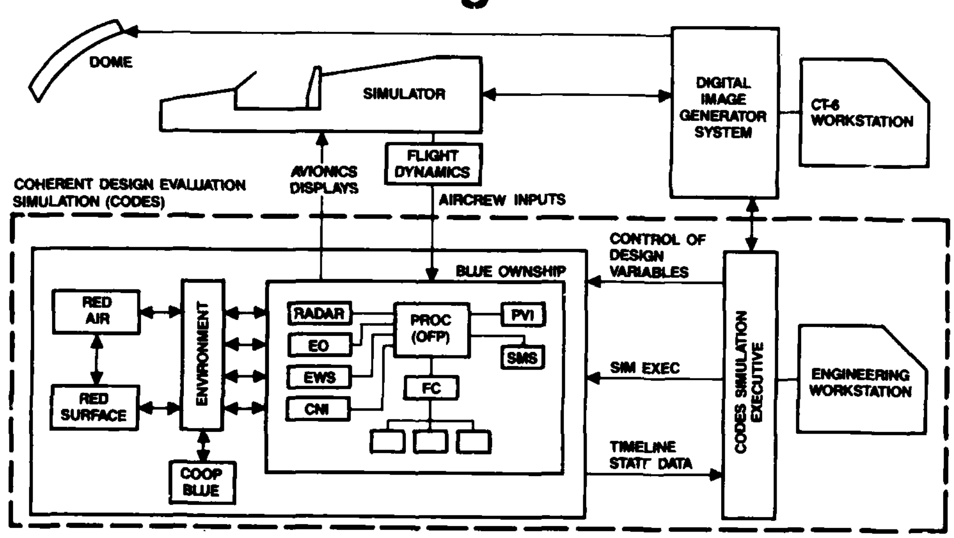
  } Mission success

(U)

FIGURE 13e (CONT'D)

(U)

Dome/CODES Integration

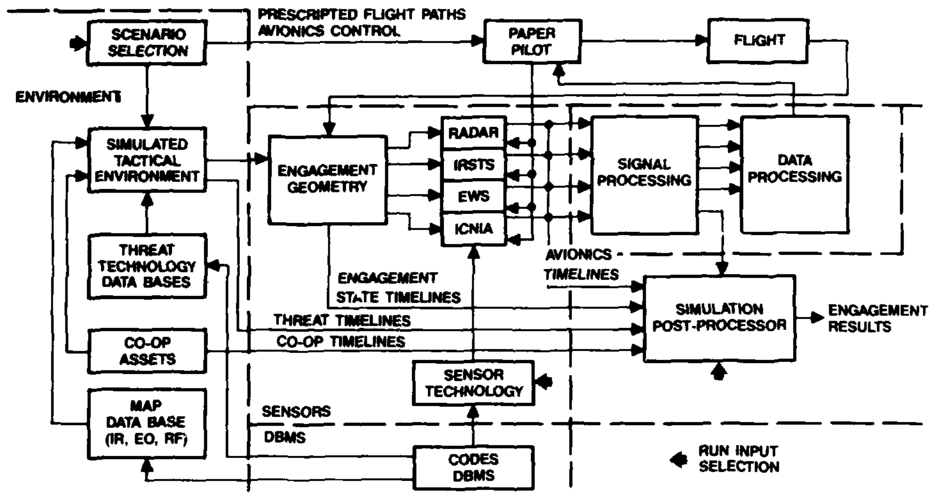


(U)

FIGURE 14

(U)

Avionics System Engineering Work Station Codes Level 1 (All Digital)



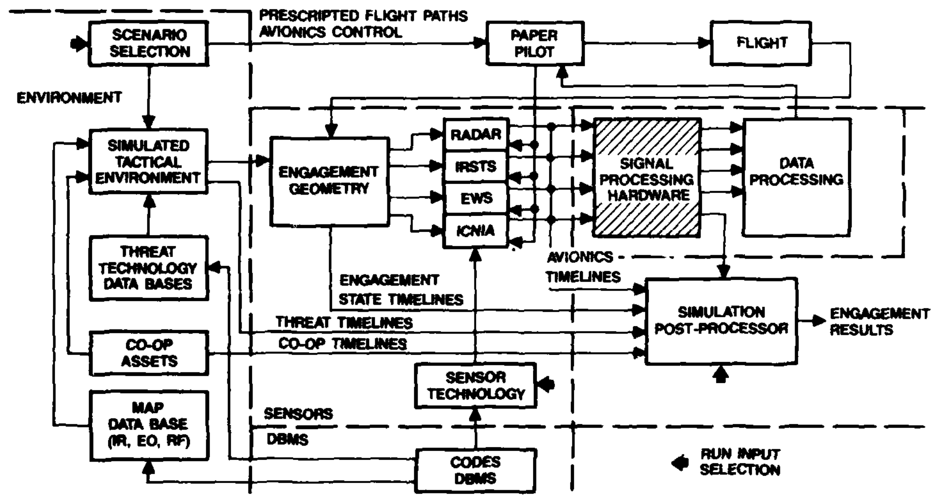
(U)

FIGURE 15

(U)

Signal Processor Hardware-in-Loop

Codes Level 2



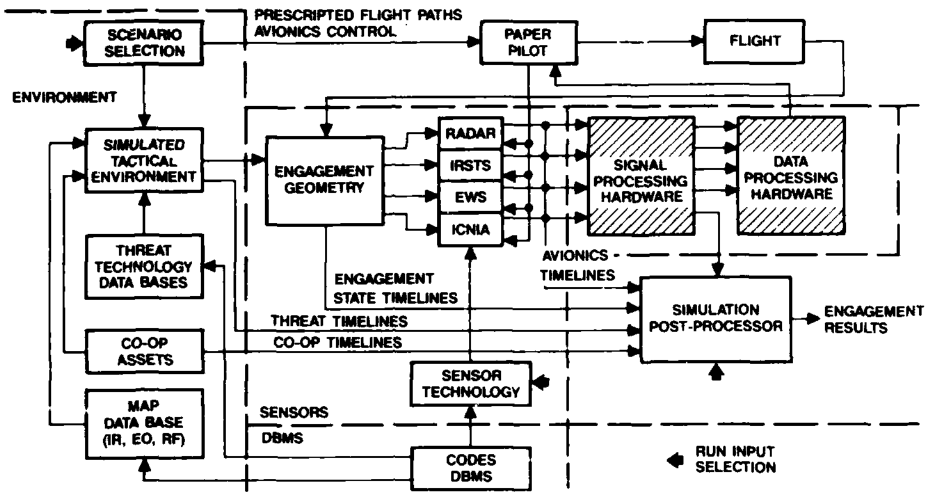
(U)

FIGURE 16

(U)

Digital Processor Hardware-in-Loop

Codes Level 3



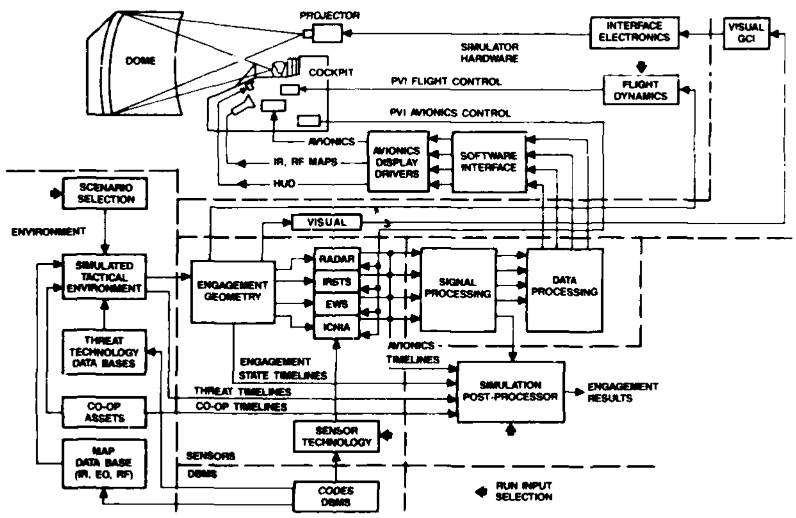
(U)

FIGURE 17

(U)

Man-in-the-Loop PVI

Codes Level 4

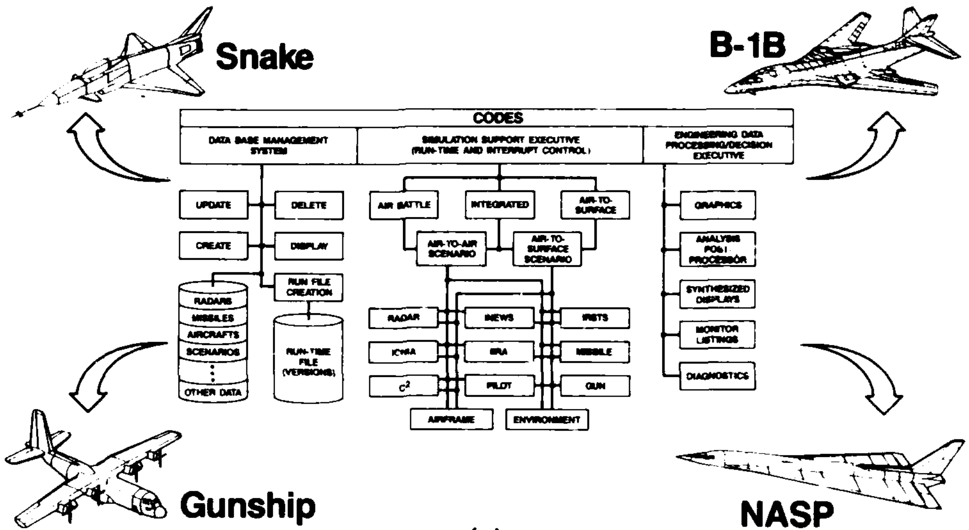


(U)

FIGURE 18

(U)

File Based – Software Reconfigurable



(U)

FIGURE 19

(U)

Summary and Conclusions

- An engineering methodology which allows detailed system, subsystem, and component design trade-offs to be performed and evaluated in an operational context IS MANDATORY, DOES NOT EXIST but IS ACHIEVABLE!
- Protracted development cycles are assured due to
  - Difficulty in achieving advocacy quickly
  - Lack of confidence in ability to meet performance predictions
  - Escalating costs
  - Inability to come to grips with evolving technology
- Concerted focus on methodology and tools by top talent with adequate funding will quickly resolve (we are solving design issues 100 times more complex than this today)
- Codes initial configuration characteristics summary
  - Hosted on two (2) Harris 1200 CPU's plus ADI-100
  - Approximately 100K LOC Fortran 77
  - Interfaced to Evens and Southerland CT-6 graphics system

(U)

FIGURE 20

A "QUASI-CONVENTIONAL" APPROACH TO FUTURE SYSTEM DESIGN AND MANAGEMENT

by
Paul A. Bross, Dipl.Phys.
Manager System Studies
Messerschmitt-Boelkow-Blohm GmbH
Postfach 801160
8000 Munich 80
Germany

Unclassified Summary

With the advent of the Bus System and the possibility for using multi-bus structures, also using distributed processing and an overall modular approach to partitioning Hardware and Software, we are now in the position to structure future Avionic Systems differently than in the past. A straightforward simple decomposition of the System into "Black Boxes", which perform only one function, will belong to the past.

My paper will deal with our approach to structuring modern integrated, multifunctional Avionic Systems and its Subsystems, making use of the new technologies emerging for system hardware and also using the new methods appearing for system design and management, however not to the vastest extent possible in terms of technologies where the advent of VHSIC and system structures decomposing down to modules would of course impose different architectures altogether.

This is why we call our approach to system architecture "quasi-conventional", because it still provides you with the impression of "black boxes". But this is not the case in terms of the past time definition.

With the arising new technologies, with digitization and computerization, our future hardware in the avionic system will be multifunctional, i.e. a single line-replaceable item LRI (equipment) will perform several functions, exhibit several modes, in many cases serving various subsystems. Nevertheless, we still use the term subsystem, actually aiding us in structuring the overall system.

The first part of my paper will be mostly devoted to outline the guidelines for designing the systems architecture, highlighting the various subsystems and "multi"-functions required, then describing the system architecture, expedite a little on bus load considerations and explain the concept of multifunctional LRIs, which serve several subsystems, provide for several functions.

This really forms a highly integrated, netted system guided by the following dialectics:

Several Subsystems	serve	one	function	or
Several LRIs	serve	one	function	
		AND		
One Subsystem	provides for	several	functions	or
One LRI	provides for	several	functions	

Designing such a system in terms of system management, system architecture, bus structure, data transfer and distributed processing will take a lot of careful investigations also looking at important issues like bus loading, computer time loading and redundancy management of system functions. The second part of my paper will outline these special design considerations on bus load and distributed processing power. It will also highlight our modern approach to a highly Integrated Test System.

In order to handle such a netted system and to manage a proper "top-down" system design, a structured approach using computer-aided tools is foreseen, also for documentation. Only with a data basis, which is established right in the beginning of the programme and which is continued consistently throughout the development phase until system integration, test and validation, a proper control can be exercised. The system decomposition during design can then be validated during integration, using the same tool and same data base.

Therefore, the third part of my paper will describe how to handle, manage and control such a complex, netted system for decomposition "top-down" as well as integration "bottom-up", test and validation, using computer-aided tools. In the absence of a

proper IPSE, we are using a tool called CORE-EPOS, whose features will be shortly described. It will be highlighted, that there is a need for integration of a test support Environment. At present the tool TUS is used but needs further development.

In conclusion it can be stated, that modern technologies available completely revise our approach to system design, system architectures and system development/management.

The hardware is dominated by the advent of Multifunction Sensors, Computerization and Digitisation which leads to the Capability of Sensorfusion and multifunctional use of Avionic items.

The software is dominated by multiprocessor items and multi media data transfer. This leads to a modular design of software, high throughput and memory and fast data transfer.

And the System Management is dominated by a set of integrated, computer aided tools, leading to a proper, consistent, effective control of all system life cycle phases.

List of Contents

-o°o-

DEVELOPPEMENT DES SYSTEMES AVIONIQUES COMPLEXES

-o-

EXPERIENCE ISSUE DES PROGRAMMES MILITAIRES FRANCAIS

-o-

PAR

ICA ANTOINE COURSIMAULT

SERVICE TECHNIQUE
DES PROGRAMMES AERONAUTIQUES - FRANCE

-o°o-

PREAMBULE

Les nouvelles générations d'aéronefs sont équipées de systèmes dont l'importance et les coûts ne cessent de croître.

Le développement de ces systèmes a nécessité la mise en place par les Services Officiels Français de méthodes :

- d'analyse du besoin
- de conception et de développement des matériels
- de production
- de suivi
- d'évolution de la définition en fonction du temps.

Ces méthodes ont essentiellement pour but d'arriver à :

- satisfaire l'utilisateur (en l'occurence les Etats-Majors)
- en respectant les délais
- en contrôlant strictement les coûts

Elles sont basées sur l'expérience des développements antérieurs et sont donc améliorées à chaque nouveau programme.

Les paragraphes qui suivent développent les idées conductrices utilisées lors du développement du MIRAGE 2000 et de l'ATL2 (ATLANTIQUE), les anomalies constatées et les améliorations susceptibles d'être apportées dans le cadre du Programme ACT (RAFALE).

I - L'ANALYSE DES BESOINS

Les besoins en renouvellement des aéronefs sont exprimés par les utilisateurs en fonction de l'évolution de leur parc et des utilisations potentielles.

Les évolutions du parc sont essentiellement dues à :

- la durée de vie
- l'attrition

Les utilisations potentielles varient selon :

- les besoins en performances :
  - longueur de piste
  - facteur de charge en combat
  - capacité d'emport
  - ...
- la définition de nouvelles missions :
  - défense aérienne
  - supériorité
  - reconnaissance très haute altitude
  - ...
- la prévision d'utilisation dans divers types de conflit :
  - centre Europe
  - théatres extérieurs
  - ...
- l'évolution des armements :
  - missiles stand off
  - armes à guidage terminal
  - ...
- l'évolution des menaces :
  - radioélectriques
  - infrarouges
  - ...

L'utilisateur a ainsi la possibilité d'exprimer ses besoins généraux, qui vont permettre de démarrer les travaux préliminaires indispensables à la bonne appréhension des besoins réels.

Il faut insister sur la difficulté de la tâche d'expression des besoins. L'Etat Major qui définit ses besoins va voir sa responsabilité engagée sur un nombre d'années très important.

Il faut en effet rappeler que le cycle de vie d'un programme d'avion s'étend sur 40 ans comme le montre le diagramme ci-dessous.

ANNEES	AVION	MOTEUR	SYSTEME ET EQUIPEMENTS	
- 10	études de base développements expérimentaux	études de base développements expérimentaux	études préparatoires développements expérimentaux	
- 5	études préparatoires	études préparatoires ----- lancement -----	1er vol maquettes Expérimentation sur maquette	
0	1er vol proto			
5	1er vol du 1er de série	-- 1er moteur de série --	phase prototype	
10			fin de mise au point	livraison série
15			rénovation	entretien majeur
20				
25				
30		Retrait du service en fonction du potentiel		

2 - CONCEPTION DES MATERIELS

A la suite de l'analyse et de la mise en forme de ces besoins généraux, va commencer le travail de conception.

Diverses solutions sont généralement capables de répondre aux besoins et la première difficulté réside dans le choix.

Celui-ci peut, bien sûr, être aidé par l'examen attentif de solutions utilisées dans certains pays en avance technologique, mais il est souvent nécessaire de se conforter par des études de base et d'effectuer des <u>développements exploratoires</u>. Les développements exploratoires peuvent s'arrêter au stade "papier" ou conduire à la fabrication de maquettes.

Le schéma suivant donne un exemple de déroulement d'un développement exploratoire ayant pour but l'étude de maquettes avionnables de Radars Aéroportés de combat aérien et d'appui sol (RACAAS).

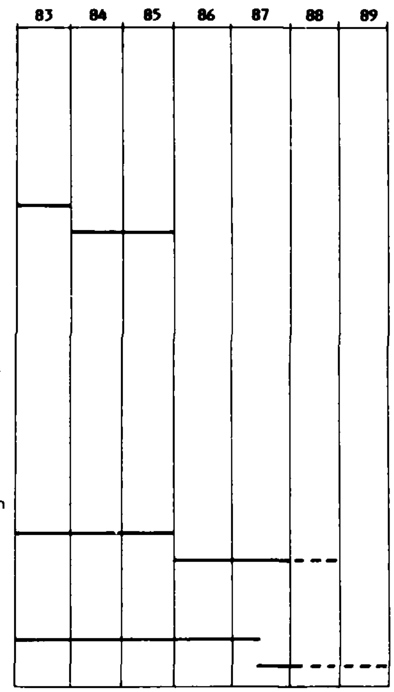
Maquette d'enregistrement <u>en vol</u> de signaux numériques reçus par un radar en moyenne et basse fréquence de récurrence
- analyse du modèle de clutter de sol
- analyse des signatures de cibles isolées ou non

Définition
Essais en vol

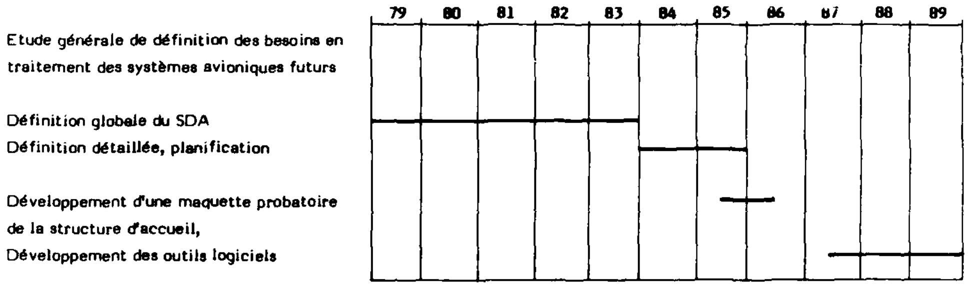
Maquette de validation des innovations pour radar futur
- moyenne fréquence de récurrence
- émetteur permettant le fonctionnement en haute, moyenne, basse fréquence de récurrence
- traitement du signal
- multicibles
- affinage doppler, image haute résolution

Définition
Essais en vol

Maquette avec balayage électronique un plan
Définition
Essais en vol



Ces développements exploratoires peuvent aussi porter sur le logiciel. Ainsi, pour doter la communauté avionique française d'un Système de Développement d'Avionique (SDA), adapté aux besoins de développement des logiciels d'équipements des systèmes embarqués, il a été décidé de lancer le développement exploratoire : ITI (intégration du traitement de l'information).



Enfin ces développements peuvent être poussés très loin comme cela a été le cas avec le démonstrateur Rafale (photo page 6), dont le rôle a été de prouver :

- le bien-fondé de la formule aérodynamique
- les qualités de commandes de vol numériques
- les possibilités qu'apportent les structures nouvelles
- la validité du concept d'interface homme/machine qui avait été regardé au cours d'un développement exploratoire sur l'organisation des postes d'équipage (OPE) (photos page 6)
    - siège incliné
    - combiné de pilotage avec tête moyenne collimatée
    - collimateur tête haute holographique
    - manche latéral
    - ...

3 - PHASE PREPARATOIRE

Les études de base et développements exploratoires ayant permis de valider des solutions technologiques, le lancement du programme peut être préparé. Les principales étapes sont :

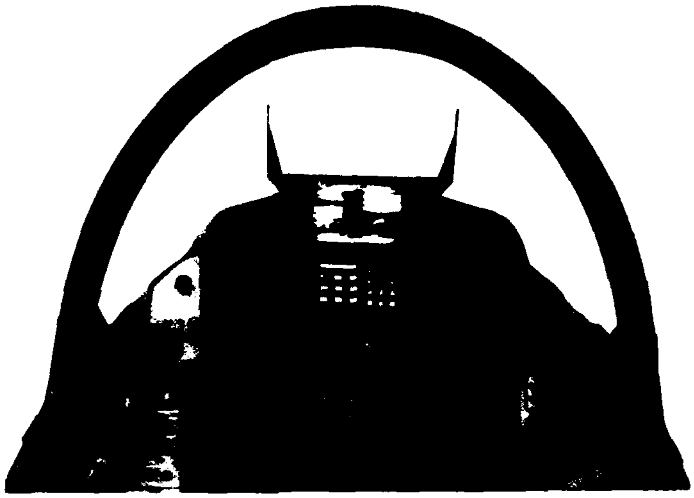
- Etablissement de la fiche programme
- Spécifications préliminaires
- Etablissement du plan de développement technique
- Etablissement du plan de financement
- Lancement

La fiche programme est l'expression du besoin des utilisateurs. Tirée des besoins généraux cités plus haut, elle précise les points les plus importants, fixe les limites du produit aussi bien vers le bas (performances minimales acceptables, ...) que vers le haut. Il faut nécessairement limiter les demandes et trouver le compromis capable d'être réalisé avec les budgets, délais, matériels disponibles.

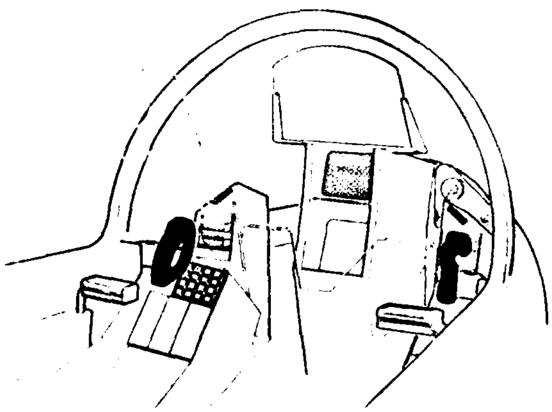
Par exemple les Etats-Majors aimeraient souvent disposer d'un avion de supériorité aérienne (grande vitesse, rapport poussée-poids élevé, grande manoeuvrabilité) et d'un avion d'appui tactique (M : 0, 9, basse altitude, capable d'emports multiples, bien protégé par des systèmes complexes de contremesures). Ils doivent se contenter d'un avion polyvalent (le compromis) pour des raisons d'économie de moyens.

Cette fiche programme, purement opérationnelle, sera transformée en spécifications préliminaires par les Services Techniques qui y ajouteront les performances à réaliser.

DEMONSTRATEUR RAFALE



POSTE D'EQUIPAGE DU RAFALE



CONCEPT D'INTERFACE HOMME/MACHINE DE L'OPE

4 - DEVELOPPEMENT

A partir des spécifications techniques, il va être possible d'entreprendre l'analyse du développement et en particulier celui du système d'armes qui sera détaillé ci-après.

La première étape consiste à effectuer l'analyse opérationnelle des besoins de façon à déterminer les solutions envisageables du point de vue :

- capteurs
- traitement et analyse du signal
- interfaces pilote
- interfaces armements
- enregistrement, transmission
- bus
- ...

Cette analyse est effectuée avec la participation de groupes de travail constitués par les principaux industriels concernés et par l'avionneur.

Elle aboutit à une architecture fonctionnelle et à une liste d'équipements.

Cette architecture, comme le montre la planche (page 8) peut être relativement complexe.

La tâche de choix de l'architecture et des équipements est du ressort des Services. Cette tâche est souvent compliquée par les nécessités de répartition entre équipementiers prenant en compte des éléments peu techniques de politique industrielle, en particulier pour les équipements dont le développement et la mise au point sont de coûts tellement élevés et l'acquisition d'expérience tellement difficile que la concurrence n'existe pas en France.

D'autre part, l'évolution des techniques vers le numérique conduisant à disposer dans chaque équipement d'un calculateur qui, par nature, sait traiter nombre de problèmes mathématiques et de gestion, il convient de valider une architecture logicielle et une répartition des calculs :

- diminuant les charges bus
- minimisant les coûts en :
  - évitant les duplications de calculateurs sophistiqués
  - réduisant au maximum les développements du logiciel

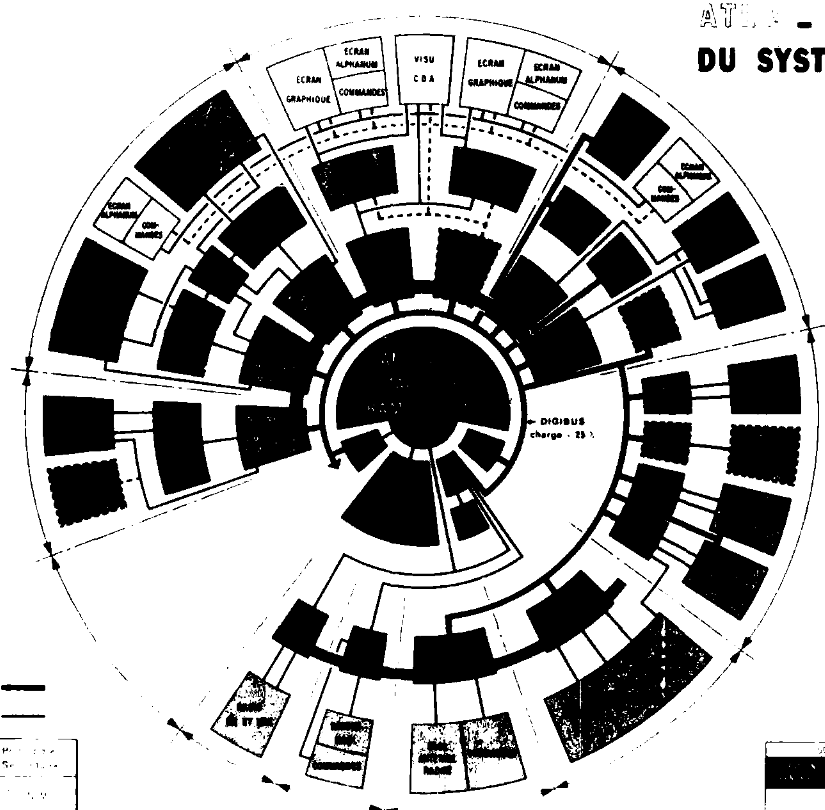
Enfin, comme cela a déjà été dit, le système sera utilisé pour de nombreuses années pendant lesquelles les évolutions technologiques seront continuelles. Il est du ressort des Services de vérifier que l'architecture choisie est susceptible de développements ou de modifications, sans problèmes majeurs.

Une des voies ouvertes consiste à essayer de créer une standardisation au niveau des entrées/sorties des équipements.

Pour chaque fonction du Système d'Armes, il faut alors définir des Spécifications Globales qui concernent l'ensemble des équipements réalisant cette fonction et qui définissent avec beaucoup de précision :

- l'objectif de la fonction
- le rôle de chacun des équipements
- les interfaces homme/machine
- les reconfigurations en cas de panne

AT[illegible] – ARCHITECTURE DU SYSTEME D'ARMES



Digibus

Liaisons numeriques serie

U.G.P

U.G.S

RAM

PROM

TRAITEMENTS SPECIFIQUES

TERMINAUX

TERMINAUX

Ce document suffit en lui-même pour définir la fonction élémentaire qui est achetée par les Services.

Bien que, actuellement, ceux-ci entrent plus dans le détail, il conviendrait, pour limiter leur charge de travail et surtout leur éviter de rentrer dans des détails du niveau du spécialiste, de limiter la surveillance des Services à l'élaboration de ces Spécifications Globales.

Afin qu'ils puissent engager leur responsabilité à ce niveau, il est nécessaire de disposer de moyens de vérification des Spécifications Globales.

Les moyens existent déjà sous la forme :

- outil d'aide à la Spécification (OASIS) (photo page 10)
- simulateur de combat du Centre Electronique de l'Armement (CELAR) (photo page 10)
- simulateur du Centre d'Essais en Vol d'Istres (C.E.V.) (photo page 10)

D'autres moyens informatiques qui pourraient être plus simples de mise en oeuvre et d'emploi que les précédents sont envisageables.

Ces moyens doivent permettre la validation et donc donner une réponse très tôt après l'édition des spécifications. En effet une réponse tardive peut donner l'effet contraire, la suite du développement s'étant poursuivi, les résultats issus des outils conduisent à des propositions de modification avec les incidences sur les prix, délais et en général bon déroulement du programme que l'on imagine.

Après validation des Spécifications Globales il serait envisageable de voir les Services acheter une fonction validée sans participer de façon détaillée à la suite du développement et en intervenant uniquement pour la réception officielle. Cet achat pourrait être forfaitaire, les prix incluant de base une provision pour correction de malfaçons éventuelles.

Seules les modifications de Spécifications Globales resteraient sous contrôle des Services. Ces modifications justifiées devraient être peu nombreuses si un consensus était obtenu au cours de la validation entre les pilotes chargés de la mise au point (pilotes constructeurs, pilotes du Centre d'Essais en Vol (C.E.V.), pilote du Centre d'Expérimentation de l'Armée de l'Air (CEAM)) et les pilotes opérationnels. Dans ce cas les vols d'essais devraient cesser d'être des vols de mise au point des Spécifications Globales pour devenir des vols de vérification finale de la Spécification, ce qui conduirait à réduire leur nombre donc leur coût. Le droit à l'erreur due à des phénomènes non accessibles en simulation (par exemple : défectuosité des capteurs, modifications des caractéristiques dues à des influences extérieures) donneraient seules lieu à des modifications justifiées.

Le développement se poursuit au niveau équipement et logiciels par l'élaboration de Spécifications détaillées.

Les équipements sont développés :

- soit sous la responsabilité de l'Etat qui en assure aussi l'intégration et le montage : matériel A
- soit sous la responsabilité des Services avec intégration par la Coordination Industrielle composée des Industriels majeurs impliqués dans le développement et de l'avionneur : matériel B
- soit directement sous la responsabilité de l'Avionneur : matériel C

La notion de Coordination Industrielle permet d'assurer la cohérence temporelle et matérielle du développement des équipements et logiciels. Elle pose par contre le problème de la concurrence industrielle et de la non-divulgation de l'expérience. L'avionneur, moins impliqué dans la concurrence que les équipementiers, sert d'agent de liaison. La Coordination Industrielle fait l'objet d'un contrat séparé.

OASIS

SIMULATEUR DE COMBAT DU CELAR

SIMULATEUR DU CENTRE D'ESSAIS EN VOL D'ISTRES

Le développement des matériels de la catégorie B peut poser plusieurs problèmes :

- non spécificité du produit qui ne pourra être adapté que par modifications soit internes, soit de la chaîne fonctionnelle
- développement asynchrone
- manque de moyens des Services pour assurer le développement d'équipements complexes

Les matériels B font l'objet de Clauses Techniques d'Intégration sous la responsabilité de l'avionneur.

L'évolution normale semble être une diminution des développements B au profit des C.

Le développement des fonctions se termine par un contrôle sur un banc d'intégration dont le but est de valider les spécifications globales à travers les équipements réels.

Avant d'être montés sur les bancs, les équipements doivent subir une recette matérielle et logicielle permettant d'éliminer une majorité de défauts. (photo page 12)

Pour cela il est nécessaire que les équipementiers puissent disposer d'un outil général de simulation de l'environnement permettant d'abord la mise au point puis donnant à la coordination industrielle ou aux Services, les moyens de recette en usine.

Cette méthode devrait aussi permettre de réduire les évolutions en phase de développement actuellement gérées par les comités locaux de modification du matériel (CLM) ou les comités de modification du logiciel (CML). Ces comités sont souvent surchargés par le nombre des évolutions, traitées sans notion d'importance et dont le bienfondé n'est pas toujours accessible aux Services chargés de les gérer du point de vue temporel et financier.

La situation actuelle est encore acceptable : il faut en effet compter 1500 modifications de tous types pour un avion dont le logiciel occupe 250K ... mais peut-on imaginer la gestion nécessaire pour un logiciel de 2500K.

Un effort considérable est nécessaire, les outils ITI doivent aider mais les outils généraux de simulation de l'environnement sont certainement nécessaires pour pouvoir traiter de tels logiciels.

Enfin cette tâche, comme il a déjà été dit, pourrait être laissée à la Coordination Industrielle, les Services se réservant les cas litigieux et ceux touchant les Spécifications globales.

Le développement se termine après introduction des dernières modifications issues des Essais en Vol.

5 - MISE EN SERIE

Le passage en série ne poserait pas de difficultés spéciales :

- si tous les développements étaient effectués de façon synchrone et la mise au point complètement terminée
- si les méthodes d'industrialisation n'avaient pas d'influence sur les performances du matériel

Le non synchronisme des développements, l'introduction de demandes nouvelles par rapport aux Spécifications Globales de base entraînent des retards de livraison des équipements et logiciels qui ont obligé à introduire la notion de standard.

BANC D'INTEGRATION DU SYSTEME D'ARMES
DU MIRAGE 2000

Les premiers standards sont livrés avec :

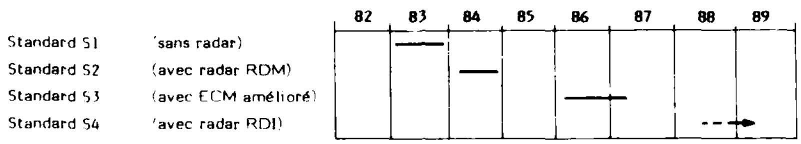
- des fonctions incomplètes ou manquantes
- des équipements encore évolutifs qui seront rattrapés par mises à hauteur successives

Il faut insister sur le fait que, si dans un premier temps, les livraisons partielles satisfont les utilisateurs, elles amènent de réels problèmes :

- de gestion du parc disponible
- de noria des équipements entraînant des diminutions de la disponibilité affectant les avions
- de reprogrammation des calculateurs et de développement indispensable d'outils de rechargement
- de mise à niveau de la documentation
- de disponibilité des moyens de test et de maintenance, eux aussi au bon standard

A ces problèmes techniques sont associés évidemment des problèmes de coût. La solution consiste à contrôler l'homogénéïté des cycles et à réduire au maximum le nombre de standards. Là encore, apparaît la nécessité du contrôle strict des Spécifications Globales et des évolutions tardives de ces spécifications.

A titre d'exemple, le diagramme ci-après donne le plan de développement et les différents standards du MIRAGE 2000.



Le standard S2 est celui correspondant aux spécifications globales initiales avec radar RDM (Radar Doppler Multifonctions)

Les méthodes d'industrialisation réagissent aussi sur le passage en série du fait :

- des modifications de méthodes de fabrication :
  - automatisation
  - changement de technologie et en particulier de composants
- des modifications de tolérances
- des changements de méthode de recette

et même parfois de la reprogrammation.

Actuellement, le développement série s'accompagne d'une phase de mise au point non négligeable. Cette phase pourrait être réduite par un contrôle plus strict de la production de série par rapport à la production prototype. Cela demande des méthodes, des moyens et du personnel très qualifié en nombre suffisant.

Cela conduit à la mise en place de la gestion "qualité".

6 - LA VIE EN UTILISATION, LES RENOVATIONS

Les diagrammes ci-avant ont montré que le début de la vie de l'avion est affectée par des restrictions opérationnelles.

Cette vie doit évoluer en fonction des demandes nouvelles des utilisateurs.

Celles-ci peuvent :

- soit être intégrées dans le système existant par un processus de modification
- soit demander une étude complexe remettant en cause le système précédemment défini. Il s'agit alors de rénovation.

Les rénovations sont des phénomènes de plus en plus courants du fait :

- de la rapidité de l'évolution de la technologie des capteurs d'adaptation
- des besoins des interfaces homme/machine
- des développements des armements et des menaces

Il est nécessaire dans la conception des systèmes de base de tenir compte dès le départ des rénovations éventuelles en créant une architecture système modulaire, capable d'extensions, de reconfigurations et travaillée au niveau des entrées et des sorties pour accepter des fonctions nouvelles.

Il faut noter qu'une telle architecture permettrait aussi en utilisation de résoudre certains problèmes d'intéropérabilité rencontrés par les utilisateurs.

Ce souhait peut voir le jour, dans les avions modernes, avec la banalisation des écrans et des commandes mais l'effort de standardisation au niveau des équipements doit être poursuivi, même si cela oblige à un droit de regard du concepteur du système sur la technologie et l'architecture interne des équipements.

CONCLUSIONS

En conclusion, les besoins suivants sont mis en évidence :

- Nécessité d'une étude et d'une formalisation précises des besoins par les utilisateurs. Ceci conditionne le bon déroulement de tout le programme

- Nécessité d'une analyse fonctionnelle, de l'écriture, de la validation et l'acceptation de Spécifications Globales des Fonctions. Ces spécifications globales constituent le niveau d'engagement entre l'Etat et les responsables du développement. Cela entraîne le besoin en outils de validation et d'acceptation des Spécifications rapides et sures

- Nécessité de réduire les Evolutions aussi bien en cours de développement prototype qu'au moment du passage en série

- Nécessité d'un contrôle constant de la qualité

L'expérience acquise sur les programmes MIRAGE 2000 pourra ainsi être reportée sur l'ACT (RAFALE) dans le but d'augmenter la qualité et les possibilités opérationnelles tout en réduisant les coûts.

-o°o-

DISCUSSION

E.Daley, UK

Emphasis was placed on the importance of agreed specifications. What form were they expressed in, and what procedures were used to check them at the specification stage?

Author's Reply

The specifications were expressed in the form of documents. The verification procedures were those described in the middle of the paper, essentially, piloted simulations as realistic as possible.

W.R.Fried, US

Would you please describe the type of communication, navigation, and radar equipment implemented on the Mirage 2000 (e.g., HF, VHF, UHF radio; voice; and data)?

Author's Reply

The Mirage 2000 is very flexible and can receive on several system variants with different types of radar: impulse doppler radar (Thomson-CSF), multimode doppler radar (Thomson-CSF), and Antilope radar (ESD). Navigation depends on an inertial system using recovery of accrued performance in the case of Antilope radar. Basically, there are two VHF/UHF radios and a transmission system for interception data (teleadhesion). Increased protection radios are available as an option.

THE FUTURE MARITIME RECONNAISSANCE AIRCRAFT AS PART OF AN INTEGRATED MARITIME BATTLEFIELD SYSTEM

by

D.Baldwinson
British Aerospace Plc
Stockport, Cheshire
UK

Although operating in a relatively more benign environment than a fast jet combat aircraft, a maritime reconnaissance aircraft is a complex weapon system which has a highly integrated set of subsystems. Future developments in the ASW/ASVW scene will strengthen the requirement for the weapon system to be a fully integrated member of a higher order system.

A move to carry out more research and development in industry, as opposed to government establishments, has helped to involve companies in higher order system investigation. This paper discusses how the concept of this greater system can be investigated, the implications for the platforms involved and the tools which could assist in these tasks.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
	AGARD-CP-417	ISBN 92-835-0437-2	UNCLASSIFIED

5. Originator	Advisory Group for Aerospace Research and Development North Atlantic Treaty Organization 7 rue Ancelle, 92200 Neuilly sur Seine, France
6. Title	THE DESIGN, DEVELOPMENT AND TESTING OF COMPLEX AVIONICS SYSTEMS
7. Presented at	the Avionics Panel Symposium held in Las Vegas, US, on 27 April—1 May 1987.

8. Author(s)/Editor(s)	9. Date
Various	December 1987

10. Author's/Editor's Address	11. Pages
Various	434

12. Distribution Statement	This document is distributed in accordance with AGARD policies and regulations, which are outlined on the Outside Back Covers of all AGARD publications.

13. Keywords/Descriptors

Avionics systems	Software
Architecture	Hardware
"Black boxes"	Interface
Sensors	VHSIC
Artificial intelligence	Sensor data fusion

14. Abstract

This symposium was designed to explore how today's system designer is addressing the solution to tomorrow's avionics systems design.

As government budgets become more limited for the research, development, testing and production of military aircraft systems, and the Warsaw Pact nations continue to produce all types of aircraft in greater numbers, the NATO nations must "re-look" at how avionic systems are developed. In general, the avionics community has thought of avionic architecture as the integration of a collection, of "black boxes" (sensor, navigation, communication, display, etc.) with the software allowing for the communication between "black boxes", computers and man. Normally, the system is decomposed into manageable parts with accurately defined interfaces. With the advent of the VHSIC, distributed processing, artificial intelligence, sensor data fusion, etc., the technologies are blurring the clear functional allocation defined for the "black box". Two factors, technology and cost, provide both an opportunity and a challenge to the system designers to design future avionics systems whose performance degrades gracefully, is reliable, and is affordable.

AGARD Conference Proceedings No.417
Advisory Group for Aerospace Research and Development, NATO
THE DESIGN, DEVELOPMENT AND TESTING OF COMPLEX AVIONICS SYSTEMS
Published December 1987
434 pages

This symposium was designed to explore how today's system designer is addressing the solution to tomorrow's avionics systems design.

As government budgets become more limited for the research, development, testing and production of military aircraft systems, and the Warsaw Pact nations continue to produce all types of aircraft in greater numbers, the

P.T.O

AGARD-CP-417

Avionics systems
Architecture
"Black boxes"
Sensors
Artificial intelligence
Software
Hardware
Interface
VHSIC
Sensor data fusion

AGARD Conference Proceedings No.417
Advisory Group for Aerospace Research and Development, NATO
THE DESIGN, DEVELOPMENT AND TESTING OF COMPLEX AVIONICS SYSTEMS
Published December 1987
434 pages

This symposium was designed to explore how today's system designer is addressing the solution to tomorrow's avionics systems design.

As government budgets become more limited for the research, development, testing and production of military aircraft systems, and the Warsaw Pact nations continue to produce all types of aircraft in greater numbers, the

P.T.O

AGARD-CP-417

Avionics systems
Architecture
"Black boxes"
Sensors
Artificial intelligence
Software
Hardware
Interface
VHSIC
Sensor data fusion

AGARD Conference Proceedings No.417
Advisory Group for Aerospace Research and Development, NATO
THE DESIGN, DEVELOPMENT AND TESTING OF COMPLEX AVIONICS SYSTEMS
Published December 1987
434 pages

This symposium was designed to explore how today's system designer is addressing the solution to tomorrow's avionics systems design.

As government budgets become more limited for the research, development, testing and production of military aircraft systems, and the Warsaw Pact nations continue to produce all types of aircraft in greater numbers, the

P.T.O

AGARD-CP-417

Avionics systems
Architecture
"Black boxes"
Sensors
Artificial intelligence
Software
Hardware
Interface
VHSIC
Sensor data fusion

AGARD Conference Proceedings No.417
Advisory Group for Aerospace Research and Development, NATO
THE DESIGN, DEVELOPMENT AND TESTING OF COMPLEX AVIONICS SYSTEMS
Published December 1987
434 pages

This symposium was designed to explore how today's system designer is addressing the solution to tomorrow's avionics systems design.

As government budgets become more limited for the research, development, testing and production of military aircraft systems, and the Warsaw Pact nations continue to produce all types of aircraft in greater numbers, the

P.T.O

AGARD-CP-417

Avionics systems
Architecture
"Black boxes"
Sensors
Artificial intelligence
Software
Hardware
Interface
VHSIC
Sensor data fusion

NATO nations must "re-look" at how avionic systems are developed. In general, the avionics community has thought of avionic architecture as the integration of a collection, of "black boxes" (sensor, navigation, communication, display, etc.) with the software allowing for the communication between "black boxes", computers and man. Normally, the system is decomposed into manageable parts with accurately defined interfaces. With the advent of the VHSIC, distributed processing, artificial intelligence, sensor data fusion, etc., the technologies are blurring the clear functional allocation defined for the "black box". Two factors, technology and cost, provide both an opportunity and a challenge to the system designers to design future avionics systems whose performance degrades gracefully, is reliable, and is affordable.

Papers presented at the Avionics Panel Symposium held in Las Vegas, US, on 27 April—1 May 1987.

ISBN 92-835-0437-2

NATO nations must "re-look" at how avionic systems are developed. In general, the avionics community has thought of avionic architecture as the integration of a collection, of "black boxes" (sensor, navigation, communication, display, etc.) with the software allowing for the communication between "black boxes", computers and man. Normally, the system is decomposed into manageable parts with accurately defined interfaces. With the advent of the VHSIC, distributed processing, artificial intelligence, sensor data fusion, etc., the technologies are blurring the clear functional allocation defined for the "black box". Two factors, technology and cost, provide both an opportunity and a challenge to the system designers to design future avionics systems whose performance degrades gracefully, is reliable, and is affordable.

Papers presented at the Avionics Panel Symposium held in Las Vegas, US, on 27 April—1 May 1987.

ISBN 92-835-0437-2

NATO nations must "re-look" at how avionic systems are developed. In general, the avionics community has thought of avionic architecture as the integration of a collection, of "black boxes" (sensor, navigation, communication, display, etc.) with the software allowing for the communication between "black boxes", computers and man. Normally, the system is decomposed into manageable parts with accurately defined interfaces. With the advent of the VHSIC, distributed processing, artificial intelligence, sensor data .fusion, etc., the technologies are blurring the clear functional allocation defined for the "black box". Two factors, technology and cost, provide both an opportunity and a challenge to the system designers to design future avionics systems whose performance degrades gracefully, is reliable, and is affordable.

Papers presented at the Avionics Panel Symposium held in Las Vegas, US, on 27 April—1 May 1987.

ISBN 92-835-0437-2

NATO nations must "re-look" at how avionic systems are developed. In general, the avionics community has thought of avionic architecture as the integration of a collection, of "black boxes" (sensor, navigation, communication, display, etc.) with the software allowing for the communication between "black boxes", computers and man. Normally, the system is decomposed into manageable parts with accurately defined interfaces. With the advent of the VHSIC, distributed processing, artificial intelligence, sensor data fusion, etc., the technologies are blurring the clear functional allocation defined for the "black box". Two factors, technology and cost, provide both an opportunity and a challenge to the system designers to design future avionics systems whose performance degrades gracefully, is reliable, and is affordable.

Papers presented at the Avionics Panel Symposium held in Las Vegas, US, on 27 April—1 May 1987.

ISBN 92-835-0437-2

AGARD

NATO OTAN

7 rue Ancelle · 92200 NEUILLY-SUR-SEINE

FRANCE

Telephone (1)47.38.57.00 · Telex 610 176

DISTRIBUTION OF UNCLASSIFIED AGARD PUBLICATIONS

AGARD does NOT hold stocks of AGARD publications at the above address for general distribution. Initial distribution of AGARD publications is made to AGARD Member Nations through the following National Distribution Centres.Further copies are sometimes available from these Centres, but if not may be purchased in Microfiche or Photocopy form from the Purchase Agencies listed below.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Coordonnateur AGARD — VSL
Etat-Major de la Force Aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Defence Scientific Information Services
Dept of National Defence
Ottawa, Ontario K1A 0K2

DENMARK
Danish Defence Research Board
Ved Idraetsparken 4
2100 Copenhagen Ø

FRANCE
O.N.E.R.A. (Direction)
29 Avenue de la Division Leclerc
92320 Châtillon

GERMANY
Fachinformationszentrum Energie,
Physik, Mathematik GmbH
Kernforschungszentrum
D-7514 Eggenstein-Leopoldshafen

GREECE
Hellenic Air Force General Staff
Research and Development Directorate
Holargos, Athens

ICELAND
Director of Aviation
c/o Flugrad
Reyjavik

ITALY
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
3 Piazzale Adenauer
00144 Roma/EUR

LUXEMBOURG
See Belgium

NETHERLANDS
Netherlands Delegation to AGARD
National Aerospace Laboratory, NLR
P.O. Box 126
2600 AC Delft

NORWAY
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
N-2007 Kjeller

PORTUGAL
Portuguese National Coordinator to AGARD
Gabinete de Estudos e Programas
CLAFA
Base de Alfragide
Alfragide
2700 Amadora

TURKEY
Milli Savunma Bakanlığı
ARGE Daire Başkanlığı
Ankara

UNITED KINGDOM
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

UNITED STATES
National Aeronautics and Space Administration (NASA)
Langley Research Center
M/S 180
Hampton, Virginia 23665

THE UNITED STATES NATIONAL DISTRIBUTION CENTRE (NASA) DOES NOT HOLD STOCKS OF AGARD PUBLICATIONS, AND APPLICATIONS FOR COPIES SHOULD BE MADE DIRECT TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS) AT THE ADDRESS BELOW.

PURCHASE AGENCIES

National Technical
Information Service (NTIS)
5285 Port Royal Road
Springfield
Virginia 22161, USA

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris, France

The British Library
Document Supply Division
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
England

Requests for microfiche or photocopies of AGARD documents should include the AGARD serial number, title, author or editor, and publication date. Requests to NTIS should include the NASA accession report number. Full bibliographical references and abstracts of AGARD publications are given in the following journals:

Scientific and Technical Aerospace Reports (STAR)
published by NASA Scientific and Technical
Information Branch
NASA Headquarters (NIT-40)
Washington D.C. 20546, USA

Government Reports Announcements (GRA)
published by the National Technical
Information Services, Springfield
Virginia 22161, USA

Printed by Specialised Printing Services Limited
40 Chigwell Lane, Loughton, Essex IG10 3TZ

ISBN 92-835-0437-2